Consultative
Committee for
Space Data Systems

DRAFT RECOMMENDATION FOR SPACE
DATA SYSTEM STANDARDS

SPACE LINK EXTENSION— RETURN VIRTUAL CHANNEL FRAMES SERVICE SPECIFICATION

CCSDS 911.2-R-1

RED BOOK

November 1997

AUTHORITY

Issue:	Red Book, Issue 1
Date:	November 1997
Location:	Oxfordshire, England

(WHEN THIS RECOMMENDATION IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS Recommendations is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This Recommendation is published and maintained by:

CCSDS Secretariat
Program Integration Division (Code MG)
National Aeronautics and Space Administration
Washington, DC 20546, USA

STATEMENT OF INTENT

(WHEN THIS RECOMMENDATION IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of member space Agencies. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommendations** and are not considered binding on any Agency.

This **Recommendation** is issued by, and represents the consensus of, the CCSDS Plenary body. Agency endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

o Whenever an Agency establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommendation**. Establishing such a **standard** does not preclude other provisions which an Agency may develop.

o Whenever an Agency establishes a CCSDS-related standard, the Agency will provide other CCSDS member Agencies with the following information:

-- The **standard** itself.

-- The anticipated date of initial operational capability.

-- The anticipated duration of operational service.

o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommendation** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommendation** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or, (3) be retired or canceled.

In those instances when a new version of a **Recommendation** is issued, existing CCSDS-related Agency standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each Agency to determine when such standards or implementations are to be modified. Each Agency is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommendation.

FOREWORD

(WHEN THIS RECOMMENDATION IS FINALIZED, IT WILL CONTAIN THE FOLLOWING FOREWORD:)

This document is a technical **Recommendation** for use in developing ground systems for space missions and has been prepared by the **Consultative Committee for Space Data Systems** (CCSDS). The Space Link Extension Return Virtual Channel Frames Service described herein is intended for missions that are cross-supported between Agencies of the CCSDS.

This **Recommendation** specifies a data service that extends certain of the space-to-ground communications services previously defined by CCSDS (references [2], [3], and [4]) within the framework established by the CCSDS Space Link Extension Reference Model (reference [1]). It allows implementing organizations within each Agency to proceed with the development of compatible, derived Standards for the ground systems that are within their cognizance. Derived Agency Standards may implement only a subset of the optional features allowed by the **Recommendation** and may incorporate features not addressed by the **Recommendation**.

Through the process of normal evolution, it is expected that expansion, deletion or modification to this document may occur. This **Recommendation** is therefore subject to CCSDS document management and change control procedures, as defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*.

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- National Aeronautics and Space Administration (NASA)/USA.
- National Space Development Agency of Japan (NASDA)/Japan.
- Russian Space Agency (RSA)/Russian Federation.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Communications Research Laboratory (CRL)/Japan.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Federal Service of Scientific, Technical & Cultural Affairs (FSST&CA)/Belgium.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Industry Canada/Communications Research Centre (CRC)/Canada.
- Institute of Space and Astronautical Science (ISAS)/Japan.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Korea Aerospace Research Institute (KARI)/Korea
- Ministry of Communications (MOC)/Israel.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Program Office (NSPO)/Taipei.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 911.2-R-1	Space Link Extension—Return Virtual Channel Frames Service Specification	November 1997	Original Issue

CONTENTS

Section Page

1 INTRODUCTION 1-1

1.1 PURPOSE OF THIS RECOMMENDATION 1-1
1.2 SCOPE 1-1
1.3 APPLICABILITY 1-1
1.4 RATIONALE 1-2
1.5 DOCUMENT STRUCTURE 1-2
1.6 DEFINITIONS, NOMENCLATURE, AND CONVENTIONS 1-3
1.7 REFERENCES 1-11

2 DESCRIPTION OF THE RETURN VIRTUAL CHANNEL FRAMES SERVICE 2-1

2.1 OVERVIEW 2-1
2.2 SPACE LINK EXTENSION REFERENCE MODEL 2-1
2.3 SERVICE MANAGEMENT 2-3
2.4 ARCHITECTURE MODEL—FUNCTIONAL VIEW 2-4
2.5 ARCHITECTURE MODEL—CROSS SUPPORT VIEW 2-6
2.6 FUNCTIONAL DESCRIPTION 2-7
2.7 OPERATIONAL SCENARIO 2-13

3 RVCF SERVICE OPERATIONS 3-1

3.1 GENERAL CONSIDERATIONS 3-1
3.2 RVCF-BIND 3-7
3.3 RVCF-UNBIND 3-12
3.4 RVCF-START 3-15
3.5 RVCF-STOP 3-20
3.6 RVCF-TRANSFER-DATA 3-22
3.7 RVCF-SYNC-NOTIFY 3-25
3.8 RVCF-ASYNC-NOTIFY 3-28
3.9 RVCF-SCHEDULE-STATUS-REPORT 3-32
3.10 RVCF-STATUS-REPORT 3-36
3.11 RVCF-GET-PARAMETER 3-38
3.12 RVCF-PEER-ABORT 3-41
3.13 RVCF-PROTOCOL-ABORT 3-43

4 RVCF PROTOCOL 4-1

4.1 GENERIC PROTOCOL CHARACTERISTICS 4-1
4.2 STATE TRANSITION TABLE 4-2

CONTENTS (continued)

Section Page

ANNEX A DATA TYPE DEFINITIONS A-1
ANNEX B INDEX TO DEFINITIONS B-1
ANNEX C ACRONYMS....................C-1
ANNEX D CONFORMANCE MATRIX D-1

Figure

2-1 Return Frame Processing SLE-FG.................... 2-4
2-2 RVCF Service Production and Provision.................... 2-6
2-3 Example of the Management and Provision of RVCF Service.................... 2-7
2-4 Simplified RVCF Service Provider State Transition Diagram.................... 2-9
2-5 Mapping of RVCF Service Operations to SLE-PDUs.................... 2-11
2-6 RVCF Service Provision Via a Gateway.................... 2-12

Table

3-1 RVCF-BIND Parameters 3-8
3-2 RVCF-UNBIND Parameters.................... 3-12
3-3 RVCF-START Parameters.................... 3-16
3-4 RVCF-STOP Parameters 3-20
3-5 RVCF-TRANSFER-DATA Parameters 3-22
3-6 RVCF-SYNC-NOTIFY Parameters 3-25
3-7 RVCF-ASYNC-NOTIFY Parameters.................... 3-28
3-8 RVCF-SCHEDULE-STATUS-REPORT Parameters.................... 3-32
3-9 RVCF-STATUS-REPORT Parameters 3-36
3-10 RVCF-GET-PARAMETER Parameters.................... 3-38
3-11 RVCF-PEER-ABORT Parameters 3-41
3-12 RVCF-PROTOCOL-ABORT Parameters 3-43
4-1 Provider Behavior 4-4
D-1 Conformance Matrix for RVCF Service (Operations)....................D-1
D-2 Conformance Matrix for RVCF Service (Other Requirements)....................D-2

1 INTRODUCTION

1.1 PURPOSE OF THIS RECOMMENDATION

The purpose of this Recommendation is to define the Space Link Extension (SLE) Return Virtual Channel Frames (RVCF) service in conformance with the SLE reference model (reference [1]). The RVCF service is an SLE transfer service that delivers to a mission user all telemetry frames from one virtual channel.

1.2 SCOPE

This Recommendation defines, in an abstract manner, the RVCF service in terms of:

a) the operations necessary to provide the service;

b) the parameter data associated with each operation;

c) the behaviors that result from the invocation of each operation; and

d) the relationship between, and the valid sequence of, the operations and resulting behaviors.

It does not specify:

a) individual implementations or products;

b) the implementation of entities or interfaces within real systems;

c) the methods or technologies required to acquire telemetry frames from signals received from a spacecraft;

d) the methods or technologies required to provide a suitable environment for communications; or

e) the management activities required to schedule, configure, and control the RVCF service.

1.3 APPLICABILITY

1.3.1 APPLICABILITY OF THIS RECOMMENDATION

This Recommendation provides a basis for the development of real systems that implement the RVCF service. Implementation of the RVCF service in a real system additionally requires the availability of a communications service to convey invocations and returns of RVCF service operations between RVCF service users and providers. This Recommendation requires that such a communications service ensure that invocations and returns of operations are transferred:

a) in sequence;

b) completely and with integrity;

c) without duplication;

d) with flow control; and

e) with notification to the application layer in the event that communications between the RVCF service user and the RVCF service provider are disrupted.

It is the specific intent of this Recommendation to define the RVCF service in a manner which is independent of any particular communications services, protocols, or technologies.

1.3.2 LIMITS OF APPLICABILITY

This Recommendation specifies the RVCF service that may be provided by an SLE System for inter-Agency cross support. It is neither a specification of, nor a design for, real systems that may be implemented for the control and monitoring of existing or future missions.

1.4 RATIONALE

The goal of this Recommendation is to increase the level of interoperability between the tracking stations or ground data handling systems of various Agencies and the consumers of spacecraft telemetry.

1.5 DOCUMENT STRUCTURE

This document is divided into four numbered sections and four annexes:

a) Section 1 presents the purpose, scope, applicability, and rationale of this Recommendation and lists the definitions, conventions, and references used throughout the Recommendation;

b) Section 2 provides an overview of the RVCF service, including a functional description, the service management context, and protocol considerations;

c) Section 3 specifies the operations of the RVCF service;

d) Section 4 specifies the dynamic behavior of the RVCF service in terms of RVCF service provider state transitions;

e) Annex A provides a formal specification of RVCF service data types using Abstract Syntax Notation One (ASN.1);

f) Annex B lists all terms used in this Recommendation and identifies where they are defined;

g) Annex C lists all acronyms used within this document;

h) Annex D provides a conformance matrix that defines what capabilities must be provided for an implementation to be considered compliant with this Recommendation.

1.6 DEFINITIONS, NOMENCLATURE, AND CONVENTIONS

1.6.1 DEFINITIONS

1.6.1.1 Definitions from OSI Basic Reference Model

This Recommendation makes use of a number of terms defined in reference [5]. The use of those terms in this Recommendation shall be understood in a generic sense; i.e., in the sense that those terms are generally applicable to technologies that provide for the exchange of information between real systems. Those terms are:

a) abstract syntax;

b) application entity;

c) application layer;

d) application process;

e) application process invocation;

f) concatenation;

g) concrete syntax;

h) flow control;

i) Open System Interconnection (OSI);

j) real system;

k) service access point (SAP);

l) transfer syntax.

1.6.1.2 Definitions from OSI Service Definition Conventions

This Recommendation makes use of a number of terms defined in reference [6]. The use of those terms in this Recommendation shall be understood in a generic sense; i.e., in the sense that those terms are generally applicable to technologies that provide for the exchange of information between real systems. Those terms are:

a) confirmation;

b) indication;

c) primitive;

d) request;

e) response.

1.6.1.3 Definitions from Abstract Syntax Notation One

This Recommendation makes use of the following terms defined in reference [7]:

a) Abstract Syntax Notation One (ASN.1);

b) object identifier;

c) (data) type;

d) (data) value.

NOTE – In Annex A of this Recommendation, ASN.1 is used for specifying the abstract syntax of RVCF service operations. The use of ASN.1 as a descriptive language is intended to support the specification of the abstract RVCF service; it is not intended to constrain implementations. In particular, there is no requirement for implementations to employ ASN.1 encoding rules. ASN.1 is simply a convenient tool for formally describing the abstract syntax of RVCF service operations.

1.6.1.4 Definitions from Telemetry Channel Coding

This Recommendation makes use of the following terms defined in reference [2]:

a) attached sync marker;

b) codeblock;

c) convolutional code;

d) pseudo-randomization;

e) Reed-Solomon check symbols;

f) Reed-Solomon code.

1.6.1.5 Definitions from Packet Telemetry

This Recommendation makes use of the following terms defined in reference [3]:

a) frame error control field;

b) master channel;

c) transfer frame.

1.6.1.6 Definitions from AOS Architecture Specification

This Recommendation makes use of the following terms defined in reference [4]:

a) coded virtual channel data unit (CVCDU);

b) cyclic redundancy code (CRC);

c) VCDU error control field;

d) virtual channel data unit (VCDU).

1.6.1.7 Definitions from SLE Reference Model

This Recommendation makes use of the following terms defined in reference [1]:

a) abstract binding;

b) abstract object;

c) abstract port;

d) abstract service;

e) initiator;

f) invoker;

g) Mission Data Operation System (MDOS);

h) Mission User Entity (MUE);

i) offline delivery mode;

j) online delivery mode;

k) operation;

l) performer;

m) physical channel;

n) responder;

o) return data;

p) Return Virtual Channel Frames channel (RVCF channel);

q) Return Virtual Channel Frames service (RVCF service);

r) service user (user);

s) service provider (provider);

t) SLE Complex;

u) SLE Complex Management;

v) SLE data channel;

w) SLE functional group (SLE-FG);

x) SLE protocol data unit (SLE-PDU);

y) SLE service data unit (SLE-SDU);

z) SLE service package;

aa) SLE System;

bb) SLE transfer service instance;

cc) SLE transfer service production;

dd) SLE transfer service provision;

ee) SLE transfer service instance provision period;

ff) SLE Utilization Management;

gg) space link;

hh) space link data channel;

ii) space link data unit (SL-DU);

jj) space link session.

1.6.1.8 Additional Definitions

1.6.1.8.1 Application Context

An application context is technology-specific information that describes both the service being offered by the provider and the required communications environment.

NOTE – For example, some OSI-based applications use an application context name and a context description that are passed to an association control service element. The TCP/IP-based Distributed Computing Environment (DCE) uses a 128-bit integer to identify interfaces. Successful negotiation of the application context must occur between user and provider before an association is established.

1.6.1.8.2 Association

An association is a cooperative relationship between an SLE service-providing application process invocation and an SLE service-using application process invocation.

1.6.1.8.3 Communications Service

A communications service is a capability that enables an SLE service-providing application process invocation and an SLE service-using application process invocation to exchange information.

NOTE – If an SLE service user and an SLE service provider are implemented using different communications services, then interoperability between them is possible only by means of a suitable gateway. Adherence to this Recommendation ensures, at least in principle, that it is possible to construct such a gateway.

1.6.1.8.4 Confirmed Operation

A confirmed operation is an operation that requires the performer to return a report of its outcome to the invoker.

1.6.1.8.5 Error Control Field

The error control field of a frame is the frame error control field (reference [3]) if the frame is a transfer frame or the VCDU error control field (reference [4]) if the frame is a VCDU or CVCDU.

1.6.1.8.6 Frame Version Number

The frame version number is either the transfer frame version number (reference [3]) or the version number in the VCDU primary header (reference [4]).

NOTE – The definitions of frame version number given in references [3] and [4] are equivalent. If a CCSDS-compatible telemetry frame is known to contain no errors, the frame version number enables one to distinguish between a transfer frame and a VCDU.

1.6.1.8.7 Invocation

The invocation of an operation is the making of a request by an object (the invoker) to another object (the performer) to carry out the operation.

1.6.1.8.8 Parameter

A parameter of an operation is data that may accompany the operation's invocation or return.

1.6.1.8.9 Performance

The performance of an operation is the carrying out of the operation by an object (the performer).

1.6.1.8.10 Port Identifier

A port identifier identifies a logical channel in a communications service. A port identifier contains all the addressing information necessary to make a connection possible.

NOTE – Port identifiers are technology-specific. For example, a port identifier might be the combination of an Internet Protocol (IP) network address and a Transport Control Protocol (TCP) port number or the combination of an Open Systems Interconnect (OSI) network address and an associated set of service access points (SAPs).

1.6.1.8.11 Return

The return of an operation is a report, from the performer to the invoker, of the outcome of the performance of the operation.

1.6.1.8.12 Spacecraft Identifier

The spacecraft identifier of a frame is as defined in reference [3] if the frame is a transfer frame or as defined in reference [4] if the frame is a VCDU or CVCDU.

1.6.1.8.13 Telemetry Frame

A telemetry frame is a transfer frame (as defined in reference [3]) or a virtual channel data unit or coded virtual channel data unit (as defined in reference [4]).

1.6.1.8.14 Unconfirmed Operation

An unconfirmed operation is an operation that does not require a report of its outcome to be returned to the invoker by the performer.

1.6.1.8.15 Virtual Channel

All telemetry frames with the same frame version number, the same spacecraft identifier, and the same virtual channel identifier on the same physical channel constitute a virtual channel.

NOTE – Depending on the frame version number, the definitions of spacecraft identifier and virtual channel identifier are as given in either reference [3] or reference [4].

1.6.1.8.16 Virtual Channel Identifier

The virtual channel identifier of a frame is as defined in reference [3] if the frame is a transfer frame or as defined in reference [4] if the frame is a VCDU or CVCDU.

1.6.2 NOMENCLATURE

The following conventions apply throughout this Recommendation:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

1.6.3 CONVENTIONS

1.6.3.1 Specification of Operations

1.6.3.1.1 General

Section 3 of this Recommendation specifies the operations that comprise the RVCF service. The specification of each operation is divided into subsections as follows:

1.6.3.1.2 Purpose Subsection

The Purpose subsection provides a brief description of the purpose of the operation. Additionally, it indicates whether the operation may be invoked by the user, provider, or both; whether the operation is confirmed or unconfirmed; and whether there are any constraints on when the operation may be invoked.

1.6.3.1.3 Invocation, Return, and Parameters Subsection

The Invocation, Return, and Parameters subsection describes the parameters associated with each operation, including their type and semantics. A table accompanying the description of each operation lists all parameters associated with the operation and, for both the invocation and return, whether the parameter is always present, always absent, or conditionally present.

For parameters that are conditionally present, the parameter description specifies the conditions for the presence or absence of the parameter. The condition is generally based on the value of another parameter in the same invocation or return; for example, the parameter `diagnostic` is present in a return if and only if the value of the parameter `result` is 'negative result'. For a conditional parameter in a return, the condition may be based on the value of a parameter in the corresponding invocation.

In the table, the following convention is used to indicate whether a parameter is always present, always absent, or conditionally present:

M	Always present
C	Conditionally present
Blank	Always absent

NOTE – Even though a parameter may be characterized as always present, its description may specify that its value is permitted to be 'null' or 'unused' or the like.

1.6.3.1.4 Effects Subsection

The Effects subsection describes the effects an operation has on the invoker, the performer, the association between them, or any combination thereof. The details of how those effects occur or the mechanisms used are outside the scope of this Recommendation.

1.6.3.2 Typographic Conventions

1.6.3.2.1 Operation Names

Names of RVCF service operations always appear in uppercase and begin with the characters 'RVCF-' (e.g., RVCF-TRANSFER-DATA).

1.6.3.2.2 Parameter Names

Names of parameters of RVCF service operations generally appear in lowercase and are always typeset in a fixed-width font (e.g., `initiator-port-identifier`).

NOTE – When the value of a parameter is referred to, rather than its name, the value is typeset in the normal text font.

1.6.3.2.3 Value and Type Names

Each parameter of each RVCF service operation is specified as being of a certain type. The type of a parameter constrains the values that may be assigned to it. Names of parameter types and values are always shown in quotation marks (e.g., ‘octet string’ and ‘no such service instance’).

NOTE – The name of a value does not imply anything about its type. For example, the value ‘no such service instance’ has the appearance of a character string but might be assigned to a parameter whose type is ‘integer’. Parameter types are specified in Annex A.

1.6.3.2.4 State Names

This Recommendation specifies the states of RVCF service providers. States may be referred to by number (e.g., state 2) or by name. State names are always shown in quotation marks (e.g., ‘active’).

1.6.3.2.5 Data Type Definitions

Data type definitions for the RVCF service are presented in Annex A in the form of a set of ASN.1 modules. Regardless of the conventions used elsewhere in this Recommendation, the text of the ASN.1 modules is typeset entirely in a fixed-width font.

1.6.3.3 Other Conventions

This Recommendation uses the conventions specified in reference [1].

1.7 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommendation are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Recommendations.

[1] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Recommendation for Space Data System Standards, CCSDS 910.4-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1996.

[2] *Telemetry Channel Coding*. Recommendation for Space Data System Standards, CCSDS 101.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, May 1992.

[3] *Packet Telemetry*. Recommendation for Space Data System Standards, CCSDS 102.0-B-4. Blue Book. Issue 4. Washington, D.C.: CCSDS, November 1995.

[4] *Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*. Recommendation for Space Data Systems Standards, CCSDS 701.0-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, November 1992.

[5] *Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. International Standard, ISO/IEC 7498-1:1994. Geneva: ISO, 1994.

[6] *Information technology—Open Systems Interconnection—Basic Reference Model—Conventions for the definition of OSI services*. International Standard, ISO/IEC 10731:1994. Geneva: ISO, 1994.

[7] *Information technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1)*. International Standard, ISO/IEC 8824:1990. 2nd ed. Geneva: ISO, 1990.

[8] *Information technology—Text Communication—Message-Oriented Text Interchange Systems (MOTIS) —Part 3: Abstract Service Definition Conventions*. International Standard, ISO/IEC 10021-3:1990. Geneva: ISO, 1990.

2 DESCRIPTION OF THE RETURN VIRTUAL CHANNEL FRAMES SERVICE

2.1 OVERVIEW

The RVCF service delivers to the user of the service all telemetry frames from one virtual channel. A virtual channel consists of all telemetry frames with the same frame version number, the same spacecraft identifier, and the same virtual channel identifier on the same physical channel. A complete specification of these concepts is provided in references [2], [3], and [4].

For delivery to the user, each frame acquired from the space link is encapsulated in an SLE service data unit (SLE-SDU) that also carries annotation (e.g., the Earth receive time of the frame). In general, the RVCF service delivers frames to the user in the order in which they were received from the space link.

The operations defined in section 3 of this Recommendation enable an RVCF service user to interact with an RVCF service provider to:

a) establish an association between the user and the provider;

b) receive annotated telemetry frames, constituting one virtual channel, from the provider;

c) obtain notifications and reports regarding the status, configuration, and performance of the service;

d) temporarily suspend and later re-start the delivery of telemetry frames; and

e) release an association.

The provision of RVCF service for one virtual channel for access by one service user constitutes one instance of service. The provision of RVCF service from one virtual channel to multiple service users and the provision of RVCF service from multiple virtual channels to one or more service users are permitted but are specified to constitute multiple service instances.

2.2 SPACE LINK EXTENSION REFERENCE MODEL

The RVCF service is specified within the framework defined by the SLE reference model (reference [1]). The SLE reference model is based, in part, on the Abstract Service Definition Conventions (ASDC) defined in reference [8]. The following paragraphs summarize selected concepts from the SLE reference model.

2.2.1 ABSTRACT OBJECT

An abstract object is a functional entity that interacts with other abstract objects. Objects are of different types, which determine their function and behavior. Objects are characterized by their interfaces, which are called abstract ports, and the operations that are made available through those interfaces (reference [1], 1.6.1.1).

2.2.2 ABSTRACT SERVICE

An abstract service is the capability provided by a set of operations that an abstract object exposes at one or more of its abstract ports (reference [1], 1.6.1.2).

NOTE – The concept of an abstract service is to be distinguished from the concept of an (N)-service as defined in the OSI Basic Reference Model (reference [5]). The definition of (N)-service is in terms of the capability provided by one layer in the OSI architecture to the layer above it. The definition of abstract service is in terms of the capability provided by one abstract object to another abstract object. In a cross support scenario where one Agency is providing an SLE service to another Agency, the object that provides the service typically is associated with one Agency, and the object that uses the service typically is associated with the other Agency.

2.2.3 SERVICE USER/PROVIDER

An object that offers a service to another by means of one or more of its ports is called a service provider (provider). The other object is called a service user (user). An object may be a provider of some services and a user of others (reference [1], 1.6.1.6).

The terms user and provider are used to distinguish the roles of two interacting objects. In this Recommendation, when two objects are involved in provision of a service, the object closer to the space link is considered to be the provider of the service, and the object further from the space link is considered to be the user.

2.2.4 ABSTRACT BINDING

When two abstract ports of the same type have an association established between them such that an abstract service can be provided, the two ports are said to be bound. The act of establishing such an association is called abstract binding (reference [1], 1.6.1.7).

The terms initiator and responder are used to describe the interaction between two objects with respect to abstract binding. One object (the initiator) invokes a bind operation which is accepted (or rejected) by another object (the responder).

2.2.5 OPERATION

An operation is a procedure or task that one object (the invoker) can request of another (the performer) through a port pair bound within the terms of an agreement (reference [1], 1.6.1.8).

The terms invoker and performer are used to describe the interaction between two objects as the operations that constitute the service occur. One object invokes an operation that is performed by the other. For most services, each object invokes some operations and performs others.

2.3 SERVICE MANAGEMENT

SLE service management determines the number, type, and schedule of RVCF service instances to be provided, the resources required to enable those service instances, and the initial configuration of all service instances and their supporting resources. SLE service management is the subject of separate CCSDS Recommendations, and the CCSDS Secretariat should be contacted for information regarding the status of those Recommendations.

The SLE reference model distinguishes between service provision and service production. Certain configuration parameters are associated with provision of RVCF service while others are associated with production. Changes to RVCF provision configuration parameters (e.g., quality of service) affect only a single service instance; the values of such parameters are initialized by service management when the service instance is created but may be modified subsequently by the user through RVCF service operations specified in this Recommendation. Changes to RVCF production configuration parameters (e.g., bit rate, frame length, coding type) potentially affect multiple service instances or potentially impact SLE Complex resources; consequently, those parameters are modified only through service management.

In general, RVCF service may be user-initiated (i.e., the user invokes the bind operation) or provider-initiated (i.e., the provider invokes the bind operation). However, any particular service instance shall support either user initiation or provider initiation but not both. Which form of initiation applies to a particular service instance is set by service management.

The RVCF service may be configured for the complete online delivery mode, the timely online delivery mode, or the offline delivery mode. Online delivery mode indicates that the provision of service is generally coincident in time with the space link session, whereas offline delivery mode indicates that the telemetry frames acquired during a space link session are provided to the user some time after the end of the space link session. Both the timely and complete online delivery modes assume the use of a reliable communications service; they differ in that the timely mode allows for the controlled discarding of telemetry frames at the application layer if it is not possible to deliver those frames within a certain amount of time after they are acquired from the space link (e.g., due to communications service backlog). Although the RVCF service supports all three delivery modes, any particular instance of service can support only one. The delivery mode applicable to a particular service instance is set by service management.

2.4 ARCHITECTURE MODEL—FUNCTIONAL VIEW

2.4.1 RETURN SPACE LINK PROCESSING FUNCTIONAL GROUP

The Return Frame Processing Functional Group (RFP-FG, shown in figure 2-1) is the SLE functional group (SLE-FG) that produces the RVCF service. As shown in the figure, the RFP-FG provides a number of different services in addition to RVCF service. This Recommendation addresses only the RVCF service; the other services are to be defined in companion Recommendations.

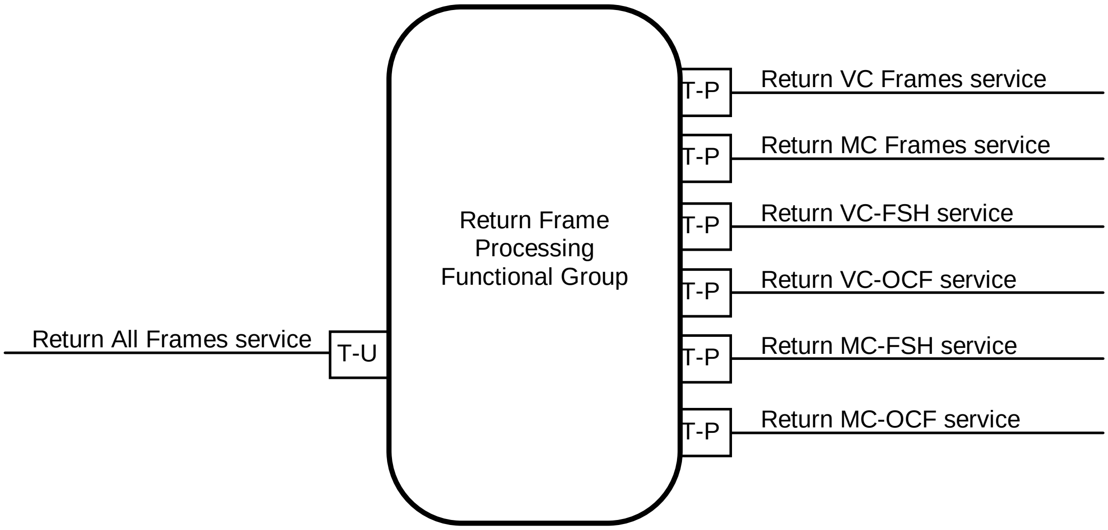

Figure 2-1: Return Frame Processing SLE-FG

As described in reference [1], the RFP-FG consumes Return All Frames (RAF) service and provides RVCF service. By means of the consumed RAF service, the RFP-FG receives an RAF channel. The RAF channel consists of a stream of SLE-SDUs that encapsulate the telemetry frames acquired from one space link physical channel. From this input, the RFP-FG produces an RVCF channel. The RVCF channel consists of a stream of SLE-SDUs. Most of these SLE-SDUs encapsulate telemetry frames from the virtual channel specified by the user of the RVCF service; such SLE-SDUs also carry annotation information associated with the frame (e.g., its Earth receive time). Other SLE-SDUs in the RVCF channel carry notification of the occurrence of certain events that may pertain to the RVCF service (e.g., loss of frame synchronization). The RVCF channel produced by the RFP-FG is delivered to a user by means of the RVCF service. More specifically, the RFP-FG performs the following functions with respect to RVCF service:

a) consumes a single RAF channel;

b) demultiplexes the RAF channel into its component virtual channels, annotates each virtual channel frame in each virtual channel to form RVCF SLE-SDUs, and injects the resulting SLE-SDUs into RVCF channels;

c) optionally, stores (and subsequently retrieves) sufficient data to reconstruct the RVCF channels for delivery through one or more offline RVCF service instances;

d) makes the RVCF channel available to online and offline RVCF service instances to effect the provision of RVCF service.

The telemetry frames delivered by the RVCF service are encapsulated in SLE-SDUs. RVCF SLE-SDUs that encapsulate telemetry frames are annotated with information that pertains to that specific frame. The annotation consists of:

a) the Earth receive time of the frame;

b) an identifier that indicates the antenna used to acquire the frame;

c) a parameter that characterizes the data link continuity of this frame with respect to the preceding frame on the space link;

d) an optional octet string that may be used to provide additional, non-standard annotations that are mutually agreed to by the SLE Complex providing the service and the Mission Data Operations System (MDOS) associated with the user of the service.

NOTES

1 The RVCF service delivers only telemetry frames that are error-free. The determination that a frame is error-free is based on the frame quality annotation provided by the RAF service: a frame is considered error-free if it was annotated by the RAF service with a frame quality of ‘good’. The RAF service annotates a frame as ‘good’ if the frame is Reed-Solomon decodable, or—if the frame is not Reed-Solomon encoded—if the frame error control field decodes successfully.

2 If the RVCF service delivers frames from a virtual channel that was Reed-Solomon encoded on the space link, the Reed-Solomon check symbols are removed from the frame before it is delivered.

3 As defined by this Recommendation, telemetry frames do not include the attached sync marker but are delimited by them.

2.4.2 RVCF SERVICE PRODUCTION AND PROVISION

One instance of RVCF service production (or, one RFP-FG instance) may be associated with multiple RVCF service instances. RVCF production is concerned with the production of RVCF channels independent of any particular instance of service. In contrast, RVCF service provision is concerned with delivering an RVCF channel to an RVCF service user. Service provision addresses such matters as when service is provided (e.g., service start and stop times), how service is provided (e.g., user-initiated or provider-initiated), and quality of service (e.g., whether the delivery mode is timely online, complete online, or offline).

The SLE-SDUs generated by RVCF service production are delivered to the service user by means of the RVCF service operations defined in section 3, which also provide additional functionality to facilitate the provision of RVCF service. In turn, the RVCF service operations are realized as SLE protocol data units (SLE-PDUs) that are exchanged between the RVCF service provider and the RVCF service user by means of an underlying communications service. The relationship between SLE-SDUs and SLE-PDUs is illustrated in figure 2-2.

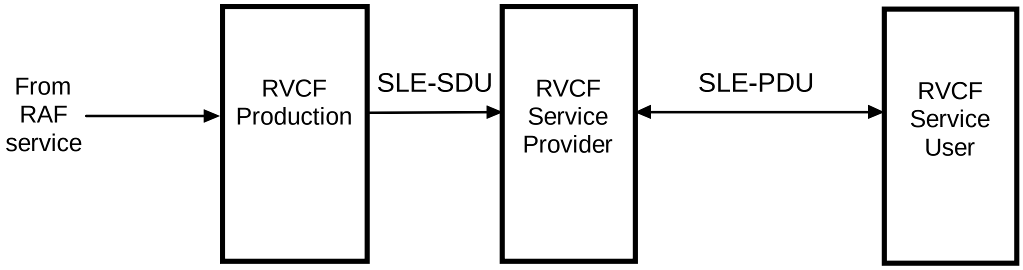

Figure 2-2: RVCF Service Production and Provision

For the online delivery mode, production and provision of the RVCF service by the provider occur concurrently with the space link session. For the offline delivery mode, service production and provision are detached, with service provision occurring some time after the end of the space link session. In the offline case, data acquired during the space link session are stored for later delivery by an offline service instance.

2.5 ARCHITECTURE MODEL—CROSS SUPPORT VIEW

The management and control of the production and provision of SLE transfer services is described in general terms in reference [1]. Figure 2-3 shows an example operational scenario and the related binding of RVCF transfer service ports (defined in this Recommendation) and the SLE management port (outside the scope of this Recommendation). This scenario shows an SLE Complex with one RFP-FG instance providing two instances of RVCF service to a Mission Data Operations System.

NOTE – Although not shown in this scenario, many other combinations are possible. For example, it is possible to have several SLE-FG instances, each consuming a different RAF channel and each providing one or more instances of service. It is also possible for the RSLP-FG and the RFP-FG to be located in different SLE Complexes.

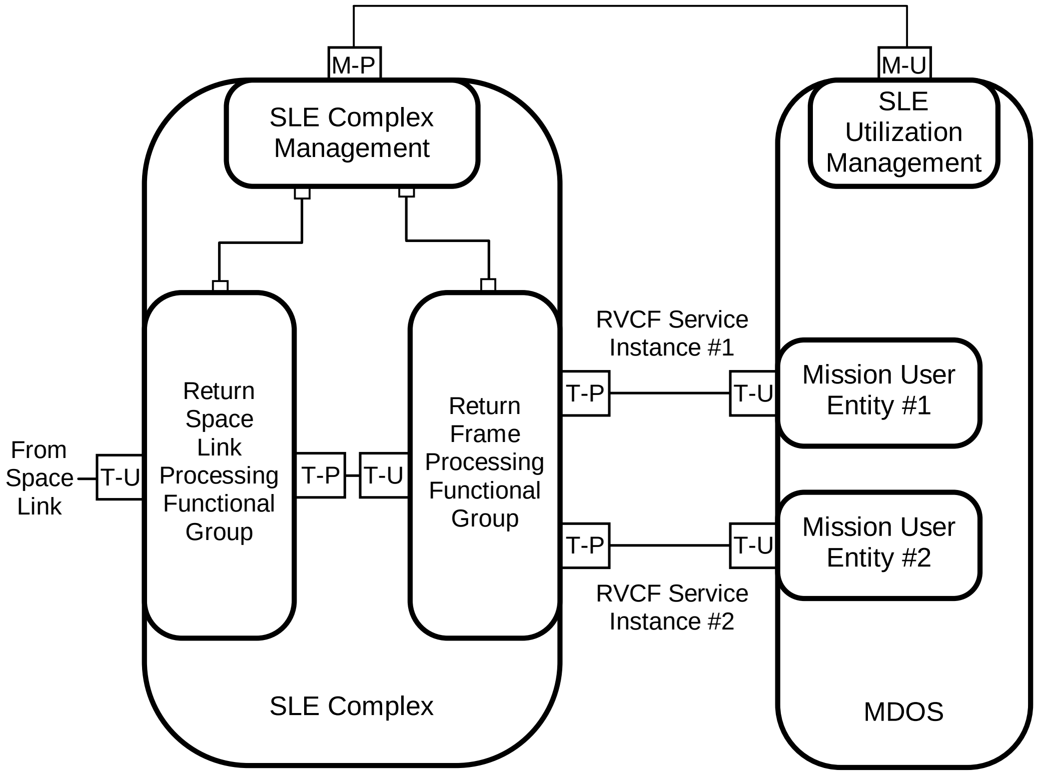

Figure 2-3: Example of the Management and Provision of RVCF Service

2.6 FUNCTIONAL DESCRIPTION

2.6.1 INTRODUCTION

This subsection describes the RVCF service with respect to scheduling, configuration, underlying services, provider states, and protocol considerations.

2.6.2 SCHEDULING AND CONFIGURATION

SLE Utilization Management negotiates with SLE Complex Management to establish mutually agreed upon SLE service packages. Among other things, SLE service packages specify what service instances are to be provided, when those services are to be provided, and what resources are needed to enable those services.

Service packages also specify the initial values of mission-dependent parameters required for service production and provision. RVCF service provision parameters include such things as the scheduled start and stop times of the RVCF service instance and the delivery mode.

Service production is guaranteed to occur only as needed to support service packages that have been scheduled and mutually agreed upon by SLE Complex Management and SLE

Utilization Management. Service provision occurs only within the bounds of the agreed upon schedule of service instances and only during those periods when there is an association between the service provider and the service user.

2.6.3 UNDERLYING SERVICES

The RVCF service is based on the functionality provided by the SLE RAF service (reference [1]) or an equivalent capability. The RAF service (or its equivalent) may be provided by the same SLE Complex that provides the RVCF service or by a different SLE Complex. Additionally, provision of RVCF service depends on service management for scheduling, resources, and configuration, and on the availability of a suitable communications service to enable the exchange of information between the RVCF service user and provider.

2.6.4 PROTOCOL DESCRIPTION

2.6.4.1 States of the Service Provider

Once a service instance is created, the RVCF service provider is in one of three states, as follows:

a) State 1 (‘unbound’): In state 1, all resources required to enable the provision of the RVCF service have been allocated, and all objects required to provide the service have been instantiated. However, no association yet exists between the user and the provider (i.e., the RVCF transfer service provider port is not bound).

b) State 2 (‘ready’): In state 2, an association has been established between user and provider, and they may interact by means of the operations described in section 3 of this Recommendation. However, the delivery of telemetry frames (by means of the RVCF-TRANSFER-DATA operation) is not permitted. The user may enable the delivery of telemetry frames by means of the appropriate service operation (RVCF-START); that, in turn, will cause the provider to transition to the state 3 (‘active’) and enable frame delivery.

c) State 3 (‘active’): State 3 is identical to state 2 except that all telemetry frames are delivered to the user as they become available. The service continues in this state until the user invokes the RVCF-STOP operation to suspend frame delivery and transition back to state 2 (e.g., in response to an end-of-data notification from the service provider signaling that the space link session has ended and all available frames have been delivered).

A simplified RVCF service provider state transition diagram is shown in figure 2-4. A detailed state transition matrix is provided in section 4.

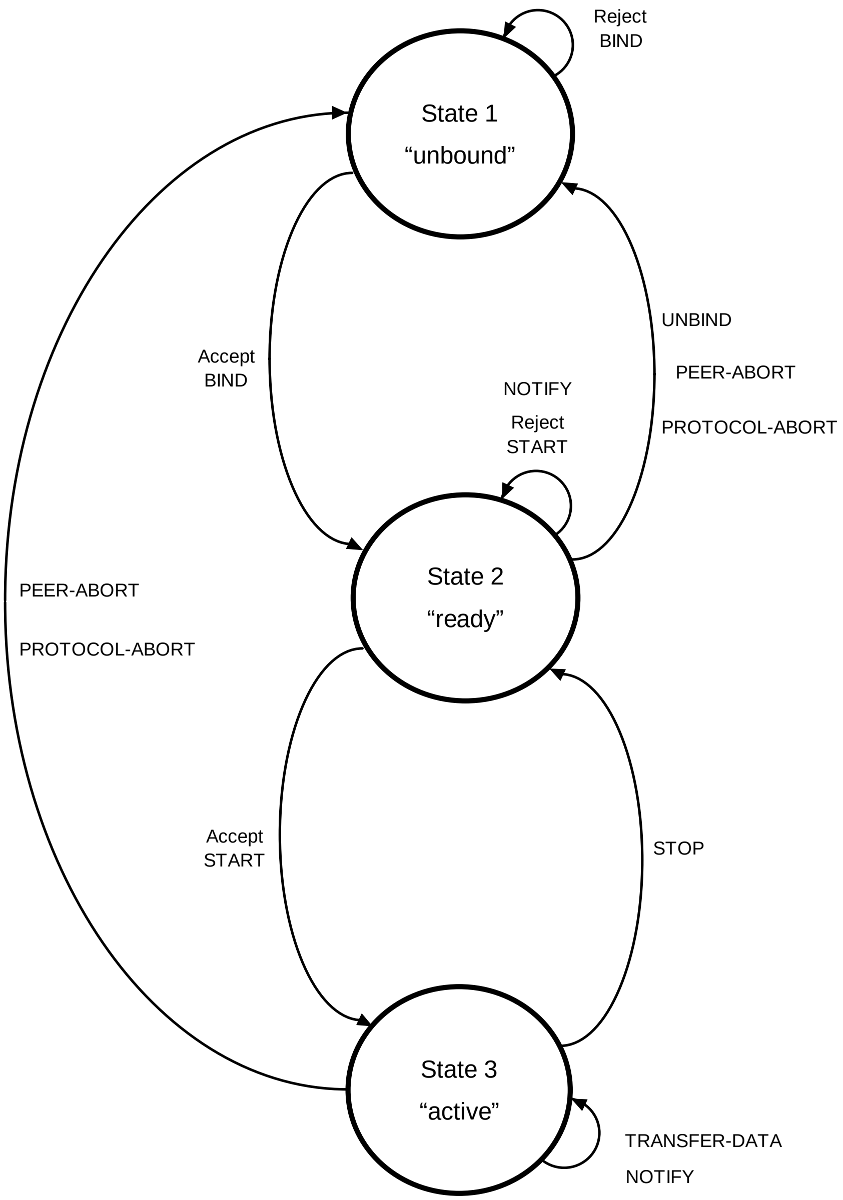

Figure 2-4: Simplified RVCF Service Provider State Transition Diagram

2.6.4.2 Termination of Association

An association is released normally when an RVCF-UNBIND operation is invoked by the initiator of the association and performed by the responder. An association may be aborted by either the user or the provider by means of the RVCF-PEER-ABORT operation. An association also may be aborted by certain failures of the communications service; such failures are signaled to the user and provider applications by the RVCF-PROTOCOL-ABORT operation.

2.6.4.3 Effects of Termination

The delivery of frames stops immediately following the release or abort of an association, and no further operations shall be exchanged between the user and the provider. The user and provider may re-establish an association via a new RVCF-BIND operation if that is consistent with the schedule for the provision of service. However, status information from the prior association is not preserved and is not available to the new association except that:

a) if the delivery mode is complete online, the contents of the online frame buffer (see 3.1.7) are persistent;

b) statistics reported by means of the RVCF-STATUS-REPORT operation (see 3.10) shall be accumulated for the entire service instance provision period.

2.6.4.4 Technology Aspects

This Recommendation defines the RVCF service. Provision of the RVCF service in a real system also requires a specification of how the RVCF service defined here is mapped to a communications service such that invocations and returns of RVCF service operations can be exchanged between the user and the provider. In order not to restrict the applicability of this Recommendation to a specific communications technology, as few assumptions as possible have been made about the characteristics of the underlying communications service (see 1.3.1).

The service interface between the user and the provider is specified in this Recommendation in terms of the operations that the service provides. However, implementation of the service in real systems requires a mapping of the service operations to protocol data units that can be exchanged by means of the communications service. Within the scope of this Recommendation, RVCF service operations are mapped to SLE-PDUs in accordance with the buffering mechanisms described in 3.1.7; e.g., multiple RVCF-TRANSFER-DATA operations may be mapped to a single SLE-PDU. From the point of view of the provider or user application, the interaction is in terms of operations; but from the point of view of the application-entities that implement the protocol, what is exchanged are SLE-PDUs that may represent multiple RVCF service operations. The mapping of service operations to SLE-PDUs is illustrated in figure 2-5.

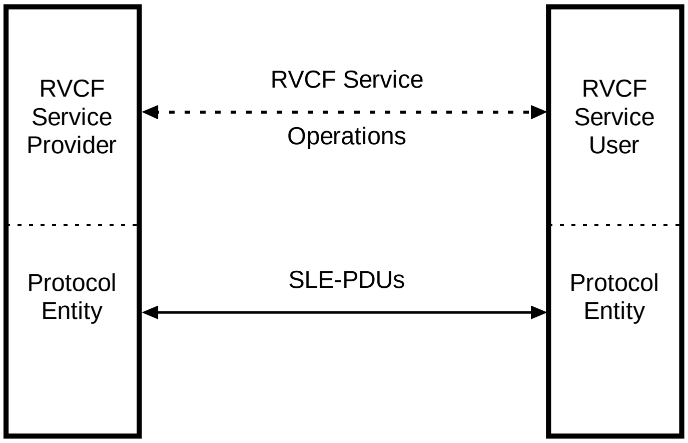

Figure 2-5: Mapping of RVCF Service Operations to SLE-PDUs

The situation is complicated further when the underlying communications service is considered. In the case where the peer applications have been built on top of different communication technologies, it is still possible to achieve interoperability, but only by means of a gateway. Figure 2-6 shows a user and a provider that use different communications technologies and interoperate through a gateway. The RVCF SLE-PDUs are translated on one side into forms appropriate for a particular communications technology. These forms are then translated by the gateway into forms appropriate for another communications technology. Finally, those forms are translated back into RVCF SLE-PDUs.

NOTE – In order to accomplish interoperability, even in the case of implementations using the same underlying communications technology, there is still the need to agree on how to map the SLE-PDUs into forms appropriate for that communications technology.

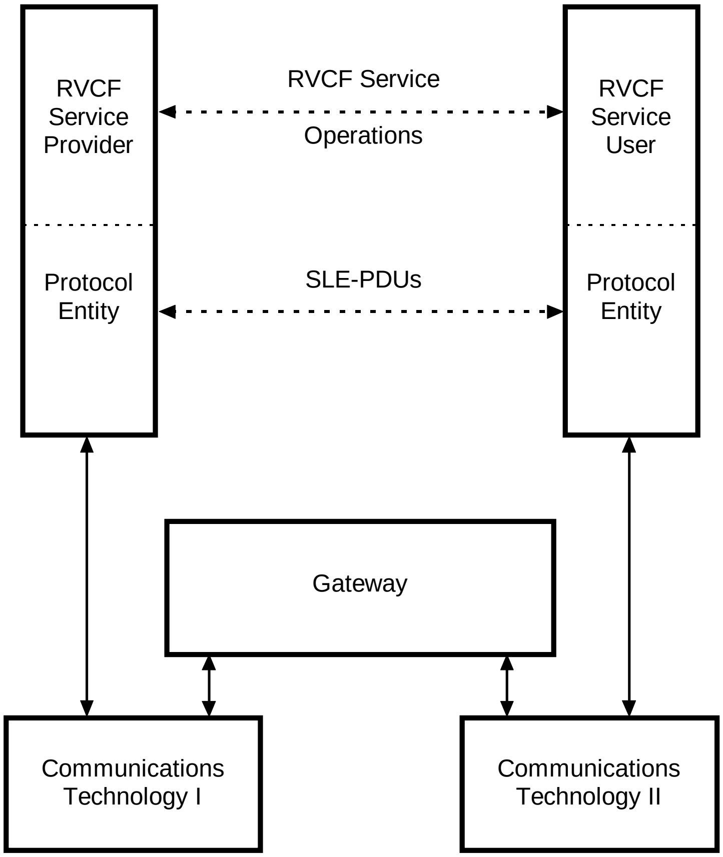

Figure 2-6: RVCF Service Provision Via a Gateway

Because the operations of the RVCF service are relatively simple, once an association is in place between the service user and the service provider, the technology specific elements involved in the exchange of SLE-PDUs are relatively minor. However, the way an association is established (i.e., the binding) tends to be specific to the communications technology in use. Nonetheless, the RVCF-BIND and RVCF-UNBIND operations as specified in this document are intended to be ‘technology neutral’. This neutrality necessitates some flexibility in the specification of the parameters of the bind operation. The semantics and even the type of some of the bind parameters may vary depending on the communications technology that is used; any such differences must be reconciled by the gateway. The following paragraphs discuss these technology-dependent parameters.

NOTE – Conceptually, it is useful to view the gateway as being part of either the SLE Complex or the MDOS. In terms of management, the parameters relevant for the binding are communicated in the form of the communications technology employed by the system which does not include the gateway. It is the responsibility of the gateway-hosting system to convert the parameters submitted by management to the ‘local’ form.

The `initiator-port-identifier` parameter identifies the port the responder shall use to communicate with the initiator. The format and content are technology dependent. A communications gateway will intercept this parameter and replace it with the port-identifier the responder will use to communicate with the gateway. Similarly, the gateway will make use of `initiator-port-identifier` to route from the responder to the initiator.

The `responder-port-identifier` parameter identifies the port to which the initiator is requesting to bind. If a gateway is used, it is actually the port of the gateway. The gateway will intercept this parameter and convert it to the port-identifier of the actual responder.

NOTE – Some technologies and/or implementations do not disclose the actual ‘physical’ address of the application, but perform a mapping by means of a directory service. This is considered a local matter and must not be visible at the level of the service specified in this Recommendation.

The `application-context` parameter identifies, in a technology specific way, the peer application. It also identifies, either explicitly or implicitly, the abstract and transfer syntaxes, that, with some technologies, are then also subject to negotiation during the RVCF-BIND operation. A gateway will need to intercept this parameter and convert it as necessary.

Binding is accomplished only when an association between the peer applications is established. A communications gateway implementation must ensure this. In particular, it must only return a report of the outcome of the RVCF-BIND operation to the initiator when it has received such a return from the responder. It is not sufficient to establish an association between the initiator and the gateway.

A gateway must also ensure that the state of the application protocol is kept consistent on the user and the provider side. In particular, in case the association with the RVCF application on one side of the gateway is aborted (e.g., because of a communications service fault), the gateway must ensure that the association with the RVCF application on the other side of the gateway is also aborted.

2.7 OPERATIONAL SCENARIO

The following illustrates a typical sequence of operations in the online delivery mode between the user and the provider of the RVCF service.

a) Prior to the actual provision of service, start and stop times for both the space link session and the associated RVCF service instance are mutually agreed upon by SLE Complex Management and SLE Utilization Management. Configuration and other information needed to enable the service are also agreed upon.

b) Some time before the scheduled start time of the RVCF service instance, the service instance is created by SLE Complex Management. Initially, the service provider is in state 1 (‘unbound’). At the scheduled start time of the space link session, the SLE Complex acquires the signal from the spacecraft and initiates the production of RVCF service. Typically (but not necessarily) the start time of the service instance will precede by a small margin the start time of the space link session to allow the user to bind to the service before the start of the space link session.

c) The user invokes the RVCF-BIND operation to establish an association.

d) The provider transitions from state 1 to state 2 (‘ready’) and returns a report of the successful completion of the bind operation to the user.

e) If the user is interested in obtaining periodic status reports, the RVCF-SCHEDULE-STATUS-REPORT operation may be invoked to configure status reporting.

f) The user invokes the RVCF-START operation to enable data flow.

g) The provider transitions from state 2 to state 3 (‘active’) and confirms the start operation to the user. As telemetry frames that meet the delivery criteria established by the user become available, they are delivered to the user by means of RVCF-TRANSFER-DATA operations. In addition, notifications may be delivered by means of RVCF-SYNC-NOTIFY and RVCF-ASYNC-NOTIFY operations, and requested status reports are delivered by means of RVCF-STATUS-REPORT operations.

h) When all available frames have been delivered, the provider will invoke the RVCF-SYNC-NOTIFY operation to provide an end-of-data notification. The end-of-data notification may be triggered, for example, because the space link session ended and all frames have been delivered or because the user specified a value for the `stop-time` parameter when invoking the RVCF-START operation.

i) By invoking the RVCF-STOP operation the user will cause the provider to transition to state 2 (‘ready’). The user may then ask for another set of frames by submitting another RVCF-START command or may terminate the association by invoking RVCF-UNBIND.

3 RVCF SERVICE OPERATIONS

3.1 GENERAL CONSIDERATIONS

3.1.1 RESULT OF OPERATIONS

3.1.1.1 All confirmed operations shall return a report of the outcome of the operation.

3.1.1.2 Except in the case of RVCF-UNBIND, all returns shall include a `result` parameter that indicates whether the outcome of the operation was successful ('positive result') or unsuccessful ('negative result').

3.1.1.3 In the event of a 'negative result', the return shall also include a diagnostic parameter that is descriptive of the reason for the 'negative result'.

NOTE – Possible values of the diagnostic parameter are listed in the description of each operation.

3.1.1.4 A diagnostic of 'other' shall be returned only if no other diagnostic in the list adequately describes the reason for the 'negative result'.

3.1.2 PARAMETER TYPES

The types of all parameters shall conform to the abstract syntax specified in Annex A.

NOTE – Some parameter types in Annex A are chosen such that possible future extensions of the range of allowed values of a parameter will not cause a type mismatch. For example, parameters that logically are of the 'enumerated' type may be specified as being of the 'named integer' type.

3.1.3 PARAMETER CHECKING

3.1.3.1 Validity checks shall be performed on the parameters of operations.

3.1.3.2 If a parameter is not valid, the operation shall not be performed, and, for confirmed operations, a report of 'negative result' shall be returned to the invoker.

3.1.3.3 Except as noted, parameter checks shall be performed in the order in which diagnostics are listed in the descriptions of the operations, and the `diagnostic` parameter shall be set to the value defined for the first problem found.

3.1.4 AUTHENTICATION

3.1.4.1 The RVCF service shall include the option to authenticate all or some of the invocations and returns of RVCF service operations within the application layer:

a) all invocations shall include an `invoker-credentials` parameter to permit the performer to authenticate the invocation;

b) all returns shall include a `performer-credentials` parameter to permit the invoker to authenticate the return;

c) the `invoker-credentials` or `performer-credentials` parameter may be set to the value ‘unused’ to signify that the invocation or return does not carry such credentials.

NOTE – Requirements for security depend on the application and the SLE system environment (e.g., whether closed or public networks are used or if access is only from physically restricted areas). In many environments, security may be provided by the communications service, transparently to the SLE application.

3.1.4.2 SLE Complex Management and SLE Utilization Management shall agree on whether or not to use the RVCF credentials parameters for authentication and, if so, for which operations.

3.1.4.3 If, by management arrangement, an invocation or return is supposed to carry credentials but does not, or the credentials it carries cannot be authenticated, then that invocation or return shall be ignored.

3.1.4.4 If, by management arrangement, an invocation or return is not supposed to carry credentials but the value of the credentials parameter is not ‘unused’, then that invocation or return shall be ignored.

3.1.4.5 The algorithms used to generate and authenticate credentials parameters must be mutually agreed upon by SLE Complex Management and SLE Utilization Management and must be known to both the RVCF service user and provider.

NOTES

1 The specification of the algorithms themselves is outside the scope of this Recommendation.

2 This Recommendation does not preclude the use of security features that are provided by the communications service or the local environment, nor does it assume the availability of such features.

3.1.5 THREADED APPLICATIONS

3.1.5.1 To support applications that may be implemented using multiple threads of execution, the parameter `invoke-ID` is specified for all confirmed operations except RVCF-BIND and RVCF-UNBIND.

NOTE – The `invoke-ID` parameter allows the invoker to correlate a particular return to the invocation that prompted it.

3.1.5.2 The `invoke-ID` parameter shall contain an invoker-supplied arbitrary integer value that shall be returned, unchanged, by the performer.

3.1.5.3 An error condition shall exist if an invocation includes an `invoke-ID` whose value is the same as that of another invocation that is awaiting confirmation.

3.1.5.4 An invoker may set the value of `invoke-ID` to ‘null’:

a) if the invoker invokes an operation with an `invoke-ID` of ‘null’, the invoker shall not invoke any other operations until the return from the current operation is received;

b) if the performer receives an invocation with an `invoke-ID` of ‘null’, the performer shall set `invoke-ID` to ‘null’ in the return.

3.1.5.5 To assure that the RVCF service behaves in a predictable manner, the effects of operations shall be as though the operations were performed in the order that their invocations were received by the performer.

NOTE – This Recommendation does not assume that applications are implemented using multiple threads of execution, nor does it preclude such implementations.

3.1.6 TIME

Universal Time Coordinated (UTC) shall be used for all parameters containing a time value.

3.1.7 DELIVERY MODES

3.1.7.1 Timely Online Delivery Mode

3.1.7.1.1 Handling of Frames

3.1.7.1.1.1 Successfully acquired frames (more precisely, the RVCF-TRANSFER-DATA invocations) shall be stored by the RVCF service provider in a buffer, called the online frame buffer, prior to their being passed to the communications service.

NOTE – When the data rate on the space link exceeds the available ground communications bandwidth or the ground communication link is congested or unavailable for a period of time, use of the complete online delivery mode (see 3.1.7.2) may lead to the accumulation of a large backlog of undelivered data resulting in the delivery of data past the point of usefulness. The timely online delivery mode provides a mechanism for ameliorating these effects by limiting the size of the backlog that is allowed to accumulate. That is accomplished by means of the buffering strategy described here.

3.1.7.1.1.2 A release timer shall be started when an RVCF-TRANSFER-DATA or RVCF-SYNC-NOTIFY invocation is stored in a buffer which was previously empty.

3.1.7.1.1.3 The duration from the time the release timer is started until it expires is given by the parameter `latency-limit`, the value of which shall be under user control.

3.1.7.1.1.4 Synchronous notifications (see 3.1.7.1.2) also shall be stored in this buffer.

3.1.7.1.1.5 The maximum size of the online frame buffer for a given instance of RVCF service shall be determined by service management, but within that limit the actual size of the buffer shall be under user control.

3.1.7.1.1.6 The contents of the buffer shall be passed to the communications service as a single unit as soon as one of the following conditions is met:

a) the buffer is full (i.e., it is not able to accommodate any more RVCF-TRANSFER-DATA or RVCF-SYNC-NOTIFY invocations); or

b) the release timer expires.

3.1.7.1.1.7 In the timely online delivery mode, the online frame buffer shall not be persistent:

a) it shall be cleared whenever the user stops data flow by invoking the RVCF-STOP operation;

b) it shall also be cleared when the association is released normally or aborted.

3.1.7.1.1.8 In the event that the buffer is unable to be passed to the communications service because of congestion or outage, the complete buffer contents shall be discarded.

NOTES

1 This buffering strategy ensures that telemetry received by the user is not older than the value given by `latency-limit` (plus some additional delay attributable to the communications service). Also, in the case of congestion on the ground communications link between service user and provider, the strategy of discarding backlogged data in large, contiguous chunks (rather than many scattered small pieces) tends to maximize the utility of the data that is delivered.

2 This strategy also tends to increase the size (and reduce the number) of discrete data units passed to the communications service, which may increase communications efficiency.

3.1.7.1.2 Handling of Notifications

3.1.7.1.2.1 Notifications delivered by means of the RVCF-SYNC-NOTIFY operation shall be buffered through the same buffer as RVCF-TRANSFER-DATA operations, and their relative ordering shall be preserved.

NOTE – The occurrence of certain events causes the service provider to issue a notification to the user. In general, there is a need for two different types of notification. For some types of events, it is desirable that the notification should be delivered synchronously with the delivery of telemetry frames; e.g., a notification of loss of frame synchronization should be delivered immediately after the last telemetry frame acquired before synchronization was lost. Notification of other events should be delivered to the user without delay, ahead of any backlog of frames awaiting delivery. The latter requirement calls for an expedited, or ‘asynchronous’, notification mechanism that bypasses the online frame buffer. In order to meet both needs, two different operations are provided.

3.1.7.1.2.2 Notifications delivered by means of the RVCF-ASYNC-NOTIFY operation shall be delivered to the user without delay, bypassing the online frame buffer.

NOTE – RVCF-TRANSFER-DATA and RVCF-SYNC-NOTIFY are the only operations that are buffered through the online frame buffer. All other operations are asynchronous, in the sense that they are always passed directly to the communications service, ahead of pending operation invocations held in the online frame buffer.

3.1.7.2 Complete Online Delivery Mode

3.1.7.2.1 For complete online delivery mode, the size of the online frame buffer shall be determined by arrangement between SLE Complex Management and SLE Utilization Management and shall be controlled through service management; it shall not be controllable by the user through the operations of the RVCF service.

NOTE – Complete online delivery mode attempts to deliver all acquired frames, in order, with minimum delay consistent with the available ground communications bandwidth. Complete online delivery requires a sufficiently large buffer to deal with communications service delays, outages, and bandwidth limitations.

3.1.7.2.2 The provider shall start to fill the online frame buffer as soon as both the service instance and the space link session have started and frames are being acquired from the space link.

3.1.7.2.3 The provider shall continue to store incoming frames in case the association is aborted.

3.1.7.2.4 Similarly, after an RVCF-STOP operation, the system shall not discard the buffer contents but shall continue to store the incoming frames.

3.1.7.2.5 In both cases, the user may, after re-binding if necessary, invoke a new RVCF-START operation, with a start time in the past, to effect delivery of the buffered data.

3.1.7.2.6 At the end of the scheduled service instance provision period, the buffer shall be released.

3.1.7.2.7 If the data rate is higher than expected or a communications outage lasts so long that the buffer overflows, frames shall be discarded from the buffer oldest first.

3.1.7.2.8 If the user, in the RVCF-START operation, requests a start time earlier than any frame still held in the buffer, the provider shall deliver the earliest data available.

3.1.7.2.9 In the interest of communications efficiency, the provider may concatenate a number of RVCF-TRANSFER-DATA and RVCF-SYNC-NOTIFY invocations into a single unit before forwarding them to the communications service:

a) the parameters determining the maximum number of invocations to be concatenated and the setting of the associated release timer shall be set by service management;

b) they shall not be controllable by the user through the operations of the RVCF service.

3.1.7.3 Offline Delivery Mode

3.1.7.3.1 Frames that are to be provided through an offline delivery mode service instance shall be acquired during the space link session and held in a buffer, called the offline frame buffer.

3.1.7.3.2 Offline service provision shall not occur until sometime after the end of the space link session.

3.1.7.3.3 During offline service provision, frames shall be retrieved from the buffer at the rate at which they are accepted by the communications service, subject to a maximum data rate limitation that may be imposed through service management.

3.1.7.3.4 The only buffering mechanism that may occur with offline delivery is the concatenation of multiple RVCF-TRANSFER-DATA invocations into a single unit for purposes of communications efficiency:

a) the parameter determining the maximum number of invocations to be concatenated shall be set by service management based on prior arrangement between SLE Complex Management and SLE Utilization Management;

b) it shall not be controllable by the user through the operations of the RVCF service.

3.2 RVCF-BIND

3.2.1 PURPOSE

3.2.1.1 The RVCF-BIND operation shall be used to establish an association between the initiator and the responder.

3.2.1.2 For every instance of RVCF service, service management shall establish whether that instance of service is to be user-initiated or provider-initiated:

a) for a user-initiated service instance, only the service user is permitted to invoke RVCF-BIND;

b) for a provider-initiated service instance, only the service provider is permitted to invoke RVCF-BIND.

3.2.1.3 The responder shall confirm the RVCF-BIND operation.

3.2.1.4 Except as provided in 3.2.1.5, the initiator of RVCF-BIND shall not invoke any other operations on this service instance until the bind is confirmed.

3.2.1.5 If the return from the invocation of RVCF-BIND is not received after a sufficiently long time, the initiator may attempt to recover by invoking RVCF-PEER-ABORT followed by another RVCF-BIND.

3.2.1.6 RVCF-BIND is valid only in state 1 (‘unbound’).

3.2.2 INVOCATION, RETURN, AND PARAMETERS

3.2.2.1 General

The parameters of the RVCF-BIND operation shall be present in the invocation and return as specified in table 3-1.

3.2.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RVCF-BIND invocation (see 3.1.4).

3.2.2.3 `performer-credentials`

The **`performer-credentials`** parameter shall provide information that enables the invoker to authenticate the return from the performance of RVCF-BIND (see 3.1.4).

Table 3-1: RVCF-BIND Parameters

Parameter	Invocation	Return
invoker-credentials	M	
performer-credentials		M
initiator-port-identifier	M	
responder-port-identifier	M	
service-type	M	
version-number	M	C
service-instance-identifier	M	
application-context	M	C
result		M
diagnostic		C

3.2.2.4 initiator-port-identifier

The **initiator-port-identifier** parameter shall identify the port the responder will use to communicate with the initiator.

3.2.2.5 responder-port-identifier

The **responder-port-identifier** parameter shall identify the port to which the initiator is requesting to bind.

NOTE – The exact format and content of initiator-port-identifier and responder-port-identifier depend on the communications technology used to make the bind (see 2.6.4.4).

3.2.2.6 service-type

The **service-type** parameter shall specify the type of service that will be provided if the bind operation succeeds.[1]

[1] For the RVCF-BIND operation, the service-type parameter is redundant, because the only valid value of service-type is ‘Return Virtual Channel Frames’. However, it is anticipated that future work by CCSDS will result in RVCF-BIND being superseded by a generic SLE-BIND operation that is invoked with any one of several SLE service types. The RVCF-BIND service-type parameter is provided in an attempt to facilitate such a change.

3.2.2.7 version-number

3.2.2.7.1 The **version-number** parameter shall identify the version number of the RVCF service specification which is to govern this association if RVCF-BIND succeeds.

3.2.2.7.2 version-number is conditionally included in the return based on the result parameter:

a) if the value of result is ‘positive result’, version-number shall be included in the return;

b) if the value of result is ‘negative result’, version-number shall not be included.

3.2.2.7.3 If included, the responder shall either:

a) accept the version proposed by the initiator by putting the same version number into the return; or

b) negotiate a lower (earlier) version number by putting the lower number into the return.

3.2.2.7.4 If the responder does not support the requested version or a lower version, it shall reject the bind with the diagnostic, ‘version not supported’.

3.2.2.7.5 If the initiator does not support the version proposed in the return, the initiator shall unbind the association.

NOTE – The version of the RVCF service defined by this issue of this Recommendation is ‘1’.

3.2.2.8 service-instance-identifier

The **service-instance-identifier** parameter shall uniquely designate this service instance within the scope of the service-providing SLE Complex.

NOTE – It is recommended that service-instance-identifier take the form of a relative distinguished name. Among other things, it should identify the SLE service agreement and the SLE service package applicable to this instance of service.

3.2.2.9 application-context

3.2.2.9.1 The **application-context** parameter shall specify technology-specific parameters required for association establishment and data transfer.

NOTE – The exact format and content depend on the communications technology used to make the bind.

3.2.2.9.2 In the RVCF-BIND invocation, `application-context` shall identify a list of eligible contexts that the invoker proposes to apply to the service session.

3.2.2.9.3 `application-context` is conditionally included in the return based on the `result` parameter:

a) if the value of `result` is ‘positive result’, `application-context` shall be included in the return;

b) if the value of `result` is ‘negative result’, `application-context` shall not be included.

3.2.2.9.4 If included, the responder shall either:

a) accept the application context proposed by the initiator by putting the same application context into the return;

b) negotiate a subset context by putting the subset context into the return.

3.2.2.9.5 If the responder does not support the proposed context or a subset of it, it shall reject the bind with the diagnostic ‘application context not supported’.

3.2.2.9.6 If the initiator does not support the context proposed in the return, the initiator shall unbind the association.

3.2.2.10 `result`

The **`result`** parameter shall specify the result of the bind operation and shall contain one of the following values:

a) ‘positive result’—the responder agrees to the bind request, and the association is established;

b) ‘negative result’ —the responder does not agree to the bind request.

3.2.2.11 `diagnostic`

3.2.2.11.1 If `result` is ‘negative result’, `diagnostic` shall be provided and shall contain one of the following values:

a) ‘no such service instance’—the requested service instance is not defined by any agreed upon service package known to the responder;

b) ‘invalid time’—the request was invoked outside the transfer service instance provision period agreed to in the service package;

c) ‘unable to comply’—the responder is unable to accept the bind at this time because of a fault affecting the responder;

d) ‘inconsistent service type’—the service type in the request is not consistent with the service type of this service instance;

e) ‘version not supported’—the responder does not support the requested version or a lower version of the service;

f) ‘application context not supported’—the requested application context is not supported by the responder;

g) ‘other’ —the reason for not binding will have to be found by other means.

NOTE – In some implementations, it may be inappropriate or impossible for the intended performer to provide a return in the event of the conditions indicated by diagnostics a) and b). Implementations should consider that, under some conditions, RVCF-BIND may fail with no return.

3.2.2.11.2 If `result` is ‘positive result’, `diagnostic` shall not be provided.

3.2.3 EFFECTS

3.2.3.1 If `result` is ‘positive result’:

a) an association between the user and the provider shall be established;

b) the provider shall transition from state 1 (‘unbound’) to state 2 (‘ready’);

c) all service parameters shall be set to the initial values prearranged by management;

d) the user may proceed to invoke other RVCF service operations to initialize the service and enable data transfer (e.g., RVCF-ASYNC-SET-PARAMETER, RVCF-SCHEDULE-STATUS-REPORT, and RVCF-START).

3.2.3.2 If `result` is ‘negative result’:

a) an association between the user and the provider shall not be established;

b) the provider shall remain in state 1 (‘unbound’);

c) the initiator should examine the `diagnostic` parameter for the cause;

d) the initiator may attempt to re-invoke the RVCF-BIND.

3.3 RVCF-UNBIND

3.3.1 PURPOSE

3.3.1.1 The RVCF-UNBIND operation shall be used to release an association previously established by RVCF-BIND.

3.3.1.2 RVCF-UNBIND shall be invoked only by the initiator (i.e., the invoker of the RVCF-BIND operation that established the association).

3.3.1.3 The responder shall confirm the RVCF-UNBIND operation.

3.3.1.4 After the initiator invokes the RVCF-UNBIND, it shall not invoke any other operations except as follows:

a) if the return from RVCF-UNBIND is not received after a sufficiently long time (to be determined by service management), the initiator shall invoke the RVCF-PEER-ABORT operation; and

b) following a successful return from RVCF-UNBIND or following the invocation of RVCF-PEER-ABORT, the initiator may issue another RVCF-BIND if permitted by the service instance schedule.

3.3.1.5 RVCF-UNBIND is valid only in state 2 (‘ready’).

3.3.2 INVOCATION, RETURN, AND PARAMETERS

3.3.2.1 General

The parameters of the RVCF-UNBIND operation shall be present in the invocation and return as specified in table 3-2.

Table 3-2: RVCF-UNBIND Parameters

Parameter	Invocation	Return
`invoker-credentials`	M	
`performer-credentials`		M
`unbind-reason`	M	

3.3.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RVCF-UNBIND invocation (see 3.1.4).

3.3.2.3 performer-credentials

The **performer-credentials** parameter shall provide information that enables the invoker to authenticate the return from the performance of RVCF-UNBIND (see 3.1.4).

3.3.2.4 unbind-reason

3.3.2.4.1 The **unbind-reason** parameter shall indicate the reason the RVCF-UNBIND operation is being invoked.

3.3.2.4.2 If the operation was invoked by the user, it shall take one of the following values:

a) 'end'—the user has obtained all frames that are needed or expected and is releasing the association normally; the provider may terminate the service instance and release its resources;

b) 'suspend'—the user is suspending service usage for an unspecified period of time; the user may re-bind to the provider to continue data transfer at any time prior to the end of the service instance provision period;

c) 'version not supported'—the user does not support the version of the RVCF service proposed by the provider in the return from RVCF-BIND; this value of unbind-reason shall be used only if the RVCF-UNBIND is the first operation invoked following the RVCF-BIND;

d) 'application context not supported'—the user does not support the application context proposed by the provider in the return from RVCF-BIND; this value of unbind-reason shall be used only if the RVCF-UNBIND is the first operation invoked following the RVCF-BIND;

e) 'other'—the reason for the release will have to be found by other means.

3.3.2.4.3 If the operation was invoked by the provider, it shall take one of the following values:

a) 'end'—the provider has transferred all available frames to the user and is releasing the association normally; the provider shall not attempt to re-bind to the user in the context of this service instance;

b) 'suspend'—the provider is suspending service provision for an unspecified period of time; the provider may attempt to re-bind to the user to continue data transfer at some time prior to the end of the service instance provision period;

c) 'version not supported'—the provider does not support the version of the RVCF service proposed by the user in the return from RVCF-BIND; this value of unbind-reason shall be used only if the RVCF-UNBIND is the first operation invoked following the RVCF-BIND;

d) ‘application context not supported’—the user does not support the application context proposed by the provider in the return from RVCF-BIND; this value of `unbind-reason` shall be used only if the RVCF-UNBIND is the first operation invoked following the RVCF-BIND;

e) ‘other’—the reason for the release will have to be found by other means.

3.3.3 EFFECTS

3.3.3.1 The association shall be released, and the initiator and the responder shall cease to communicate with each other.

3.3.3.2 The provider shall transition to state 1 (‘unbound’).

3.3.3.3 If `unbind-reason` is ‘end’, the provider may terminate the service instance and release its resources.

3.3.3.4 If `unbind-reason` is not ‘end’, the initiator may attempt to re-bind at any time prior to the end of the service instance provision period.

NOTES

1 The responder must perform the RVCF-UNBIND operation so long as it was invoked by the initiator in state 2 (‘ready’). Because there is no reason for a ‘negative result’, there is no `result` parameter in the return.

2 The performance of RVCF-UNBIND for a particular service instance does not necessarily terminate RVCF production; e.g., if `unbind-reason` is not ‘end’ and the delivery mode is complete online or if another service instance is dependent on the production.

3.4 RVCF-START

3.4.1 PURPOSE

3.4.1.1 The user shall invoke the RVCF-START operation to request that the provider begin the delivery of telemetry frames.

3.4.1.2 The provider shall confirm the RVCF-START operation.

3.4.1.3 Following a successful RVCF-START, the provider shall deliver telemetry frames acquired from the space link to the user as quickly as those frames are available.

NOTE – Communications service delays will affect the rate at which available frames are delivered.

3.4.1.4 Frames shall be delivered by means of RVCF-TRANSFER-DATA operations (see 3.6).

3.4.1.5 All frames delivered following the RVCF-START but prior to the next RVCF-STOP (see 3.5) shall be delivered in the order in which they were received from the space link.

3.4.1.6 The user may specify, as parameters of the RVCF-START, the ERTs of the first and last telemetry frames that are to be delivered by the provider.

3.4.1.7 The time parameters may be changed during an association by invoking an RVCF-STOP followed by an RVCF-START with new time parameters.

NOTE – This capability is intended primarily to support the offline delivery mode.

3.4.1.8 RVCF-START is valid only in state 2 (‘ready’) and shall be invoked only by the user.

3.4.2 INVOCATION, RETURN, AND PARAMETERS

3.4.2.1 General

The parameters of the RVCF-START operation shall be present in the invocation and return as specified in table 3-3.

3.4.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RVCF-START invocation (see 3.1.4).

3.4.2.3 performer-credentials

The **performer-credentials** parameter shall provide information that enables the invoker to authenticate the return from the performance of RVCF-START (see 3.1.4).

3.4.2.4 invoke-ID

The RVCF service provider shall return unchanged the user-supplied value of the invoke-ID parameter (see 3.1.5).

Table 3-3: RVCF-START Parameters

Parameter	Invocation	Return
invoker-credentials	M	
performer-credentials		M
invoke-ID	M	M
start-time	M	
stop-time	M	
global-VCID	M	
result		M
diagnostic		C

3.4.2.5 start-time

3.4.2.5.1 The **start-time** parameter shall contain a time value whose presence indicates that only frames with an ERT equal to or later than start-time shall be delivered.

3.4.2.5.2 For the online delivery modes, only frames acquired during the space link session associated with this service instance shall be delivered, regardless of the value of start-time.

3.4.2.5.3 For the offline delivery mode, the provider shall deliver all available frames that meet the delivery criteria regardless of the space link session in which they were acquired.

3.4.2.5.4 For the online delivery modes, if start-time is ‘null’, the data transfer shall begin with the next frame that is acquired from the space link.

3.4.2.5.5 For the offline delivery mode, start-time must not be ‘null’.

3.4.2.5.6 To be valid, start-time must satisfy the following criteria:

a) for the online delivery modes, `start-time` must be equal to or later than the start time of the service instance provision period for this service instance;

b) for the online delivery modes, `start-time` must be earlier than the end time of the service instance provision period for this service instance;

c) `start-time` must be earlier than `stop-time` (see 3.4.2.6).

NOTE – The provider is able to deliver only frames that have been acquired from the space link. For example, in an online service instance, if `start-time` is earlier than the start time of the space link session, the first frame that is delivered will be the first frame acquired after the start of the space link session.

3.4.2.6 stop-time

3.4.2.6.1 The **`stop-time`** parameter shall contain a time value whose presence indicates that data transfer should cease when the next frame that would be transferred has an ERT later than `stop-time`.

3.4.2.6.2 For the online delivery modes, only frames acquired during the space link session associated with this service instance shall be delivered, regardless of `stop-time`.

3.4.2.6.3 For the offline delivery mode, the provider shall deliver all available frames that meet the delivery criteria regardless of the space link session in which they were acquired.

3.4.2.6.4 For the online delivery modes, if `stop-time` is ‘null’, the provider shall continue to transfer all frames that are acquired from the space link and satisfy the delivery criteria until either the user invokes an RVCF-STOP operation or the association is released or aborted.

3.4.2.6.5 For the offline delivery mode, `stop-time` must not be ‘null’.

3.4.2.6.6 To be valid, `stop-time` must satisfy the following criteria:

a) `stop-time` must be later than the `start-time` (see 3.4.2.5);

b) for the online delivery modes, `stop-time`, if not ‘null’, must be earlier than or equal to the end time of the service instance provision period for this service instance;

c) for the offline delivery mode, `stop-time` plus the offline processing latency must be earlier than the current time.

NOTES

1 Offline processing latency is the length of time after a frame is acquired from the space link before it is available for retrieval using the offline delivery mode. The actual value of offline processing latency is negotiated between SLE Complex Management and SLE Utilization Management.

2 Offline delivery is only available for frames which already have been acquired when the RVCF-START operation is invoked.

3.4.2.7 `global-VCID`

The **`global-VCID`** parameter shall identify the virtual channel that is to be provided to the user and shall contain the concatenated frame version number, spacecraft identifier, and virtual channel identifier.

NOTES

1 The definitions of spacecraft identifier and virtual channel identifier depend on the frame version number. If the frame version number indicates that the virtual channel consists of transfer frames, then the definitions of spacecraft identifier and virtual channel identifier are as per reference [3]. If the frame version number indicates that the virtual channel consists of VCDUs or CVDCUs, then the definitions of spacecraft identifier and virtual channel identifier are as per reference [4].

2 The physical channel is not specified directly through the RVCF service. Rather, the selection of physical channel is determined through the service package, which specifies the RAF service instance that is consumed by the RFP-FG that is producing the RVCF service.

3.4.2.8 `result`

The **`result`** parameter shall specify the result of the RVCF-START operation and shall contain one of the following values:

a) ‘positive result’—the RVCF-START operation is accepted by the provider, and the provider shall thenceforth invoke RVCF-TRANSFER-DATA operations as needed to transfer to the user all available frames that meet the specified timeliness and data quality criteria;

b) ‘negative result’—the RVCF-START operation is not accepted by the provider, and the provider shall not invoke any RVCF-TRANSFER-DATA operations even if frames are available.

3.4.2.9 `diagnostic`

3.4.2.9.1 If `result` is ‘negative result’, `diagnostic` shall be provided and shall contain one of the following values:

a) ‘duplicate Invoke-ID’—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) ‘unable to comply’—the provider is unable to transfer data at this time because of a fault affecting the provider;

c) ‘invalid start time’—the value of the `start-time` provided in the invocation is not valid;

d) ‘invalid stop time’—the value of the `stop-time` provided in the invocation is not valid;

e) ‘missing time value’—for the offline delivery mode, the value of `start-time` and/or `stop-time` was ‘null’;

f) ‘invalid global-VCID’—the value of `global-VCID` is invalid;

g) ‘other’—the reason for the negative result will have to be found by other means.

3.4.2.9.2 If `result` is ‘positive result’, `diagnostic` shall not be provided.

3.4.3 EFFECTS

3.4.3.1 If `result` is ‘positive result’:

a) the provider shall transition to state 3 (‘active’);

b) in the ‘active’ state, the provider shall transfer frames to the user whenever they are available and satisfy the delivery criteria.

3.4.3.2 If `result` is ‘negative result’:

a) the provider shall remain in state 2 (‘ready’) and shall not deliver frames even if they are available;

b) if the diagnostic is ‘unable to comply’,

   1) the user may re-invoke the RVCF-START operation at a later time;

   2) if the provider’s being ‘unable to comply’ is more than a transient problem, the provider may invoke the RVCF-PEER-ABORT operation in response to the re-invocation of RVCF-START.

3.5 RVCF-STOP

3.5.1 PURPOSE

3.5.1.1 The user shall invoke the RVCF-STOP operation to request that the provider stop delivering telemetry frames.

NOTE – The user may re-enable frame delivery by invoking the RVCF-START operation.

3.5.1.2 The provider shall confirm the RVCF-STOP operation.

3.5.1.3 RVCF-STOP is valid only in state 3 (‘active’) and shall be invoked only by the user.

3.5.2 INVOCATION, RETURN, AND PARAMETERS

3.5.2.1 General

The parameters of the RVCF-STOP operation shall be present in the invocation and return as specified in table 3-4.

Table 3-4: RVCF-STOP Parameters

Parameters	Invocation	Return
invoker-credentials	M	
performer-credentials		M
invoke-ID	M	M
result		M
diagnostic		C

3.5.2.2 invoker-credentials

The **invoker-credentials** parameter shall provide information that enables the performer to authenticate the RVCF-STOP invocation (see 3.1.4).

3.5.2.3 performer-credentials

The **performer-credentials** parameter shall provide information that enables the invoker to authenticate the return from the performance of RVCF-STOP (see 3.1.4).

3.5.2.4 `invoke-ID`

The RVCF service provider shall return unchanged the user-supplied value of the `invoke-ID` parameter (see 3.1.5).

3.5.2.5 `result`

The **`result`** parameter shall specify the result of the RVCF-STOP operation and shall contain one of the following values:

a) ‘positive result’—the RVCF-STOP operation is accepted by the provider, and the delivery of telemetry frames to the user ceases;

b) ‘negative result’—the RVCF-STOP operation is rejected by the provider, and the delivery of telemetry frames to the user continues.

3.5.2.6 `diagnostic`

3.5.2.6.1 If `result` is ‘negative result’, `diagnostic` shall be provided and shall contain one of the following values:

a) ‘duplicate Invoke-ID’—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) ‘other’—the reason for the negative result will have to be found by other means.

3.5.2.6.2 If `result` is ‘positive result’, `diagnostic` shall not be provided.

3.5.3 EFFECTS

On performing the RVCF-STOP operation, the provider shall:

a) cease invoking RVCF-TRANSFER-DATA operations;

b) discard the contents of the online frame buffer if the delivery mode is timely online;

c) deliver an ‘end of data’ notification to the user;

d) transition to state 2 (‘ready’).

3.6 RVCF-TRANSFER-DATA

3.6.1 PURPOSE

3.6.1.1 The provider shall invoke the RVCF-TRANSFER-DATA operation to deliver a telemetry frame to the user.

3.6.1.2 RVCF-TRANSFER-DATA is valid only in state 3 (‘active’) and shall be invoked only by the provider.

NOTE – Although RVCF-TRANSFER-DATA is an unconfirmed operation, it is assumed that the communications service provides certain guarantees, as described in 1.3.1.

3.6.2 INVOCATION, RETURN, AND PARAMETERS

3.6.2.1 General

The parameters of the RVCF-TRANSFER-DATA operation shall be present in the invocation and return as specified in table 3-5.

Table 3-5: RVCF-TRANSFER-DATA Parameters

Parameters	Invocation
`invoker-credentials`	M
`earth-receive-time`	M
`antenna-ID`	M
`data-link-continuity`	M
`private-annotation`	M
`data`	M

3.6.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the user to authenticate the RVCF-TRANSFER-DATA invocation (see 3.1.4).

3.6.2.3 `earth-receive-time`

3.6.2.3.1 The **`earth-receive-time`** parameter shall contain the UTC time at which the signal event corresponding to the leading edge of the first bit of the telemetry frame was presented at the phase center of the antenna used to acquire the frame.

3.6.2.3.2 The first bit of the frame is the first bit following the attached sync marker.

3.6.2.4 antenna-ID

3.6.2.4.1 The **antenna-ID** parameter shall indicate which antenna of the SLE Complex was used to acquire this frame of data.

NOTE – antenna-ID is provided specifically to identify the physical location used as the reference point for the earth-receive-time parameter.

3.6.2.4.2 SLE Complex Management and SLE Utilization Management shall mutually agree upon the allowable values for antenna-ID and their interpretation.

NOTE – It is assumed that antenna-ID is an identifier that may be used to look up the actual location information, which is provided outside the scope of this service.

3.6.2.5 data-link-continuity

3.6.2.5.1 The **data-link-continuity** parameter shall indicate whether this frame was the direct successor on the space link of the previous frame acquired by the instance of RAF production associated with this service instance.

3.6.2.5.2 The **data-link-continuity** parameter shall contain an integer value:

a) A value of '-1' shall indicate that this is the first frame after the start of production;

b) a value of '0' shall indicate that this frame is the direct successor to the last frame;

c) any non-zero positive value shall indicate that this frame is not the direct successor to the last frame acquired from the space link:

   1) a non-zero positive value further indicates an estimate of the number of frames that were missed since the last frame acquired before this frame;

   2) a value of '1' may be used if no better estimate is available.

3.6.2.6 private-annotation

The **private-annotation** parameter shall be used to convey additional information that may be associated with a frame:

a) it may be set to 'null' to indicate that there is no private annotation;

b) if not 'null', there must be a prior arrangement between SLE Complex Management and SLE Utilization Management regarding the contents and interpretation of this parameter.

3.6.2.7 `data`

The **`data`** parameter shall contain the telemetry frame acquired from the virtual channel that is being delivered by the provider to the user; depending on the configuration of the space link, the frame may be a transfer frame, a VCDU, or a CVCDU with the Reed-Solomon check symbols removed.

NOTE – The attached sync marker is not delivered as part of `data`.

3.6.3 EFFECTS

3.6.3.1 A telemetry frame acquired by the provider from the virtual channel shall be delivered to the user.

3.6.3.2 The state of the provider shall remain unchanged.

3.7 RVCF-SYNC-NOTIFY

3.7.1 PURPOSE

3.7.1.1 The RVCF service provider shall invoke the RVCF-SYNC-NOTIFY operation to notify the user of the occurrence of an event.

NOTES

1 Notification of events may be of value to the user in understanding specific provider behavior, such as an interruption in frame delivery.

2 The order in which RVCF-SYNC-NOTIFY and RVCF-TRANSFER-DATA invocations are delivered to the user is significant. For example, if an RVCF-SYNC-NOTIFY invocation is delivered after one RVCF-TRANSFER-DATA invocation but before another, then the event indicated by the notification occurred after the ERT of the frame associated with the preceding RVCF-TRANSFER-DATA but before the ERT of the frame associated with the following RVCF-TRANSFER-DATA.

3 RVCF-SYNC-NOTIFY is an unconfirmed operation.

3.7.1.2 RVCF-SYNC-NOTIFY is valid in state 2 (‘ready’) and state 3 (‘active’) and shall be invoked only by the provider.

NOTE – Some notifications are generated only in state 3, as described below.

3.7.2 INVOCATION, RETURN, AND PARAMETERS

3.7.2.1 General

The parameters of the RVCF-SYNC-NOTIFY operation shall be present in the invocation and return as specified in table 3-6.

Table 3-6: RVCF-SYNC-NOTIFY Parameters

Parameter	Invocation
`invoker-credentials`	M
`notification-type`	M
`notification-value`	C

3.7.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the user to authenticate the RVCF-TRANSFER-DATA invocation (see 3.1.4).

3.7.2.3 `notification-type`

The **`notification-type`** parameter shall describes the event and shall contain one of the following values:

a) 'loss of frame synchronization'—the delivery of RVCF frames may be interrupted because the frame synchronization process is not able to synchronize to the stream of frames from the space link:

NOTE – Because this notification refers to processing of frames from the space link, it may or may not indicate that frames were lost on this virtual channel.

1) the notification shall be invoked once and only once for each occurrence of a frame synchronizer transition from 'in-lock' to 'out-of-lock';

2) the provider shall minimize the latency from the time the loss of frame synchronization event occurs until the notification is invoked;

3) there shall be no explicit notification when the frame synchronizer transitions from 'out-of-lock' back to 'in-lock'; rather, the next invocation of RVCF-TRANSFER-DATA shall implicitly convey the occurrence of that event;

4) loss of frame synchronization notifications shall be invoked only in state 3 and only if the delivery mode is timely online or complete online;

b) 'data discarded due to excessive backlog'—the provider discarded the contents of the online frame buffer due to timeliness considerations (see 3.1.7.1):

1) this notification shall be the first invocation stored in the online frame buffer following the discarding of the previous contents;

2) if the contents of the online frame buffer are discarded two or more times in a row without a successful intervening delivery of frames to the user, no more than one data discarded notification shall be delivered to the user;

3) data discarded notifications shall be invoked only in state 3 and only if the delivery mode is timely online;

c) 'end of data'—the provider has no more data to send: 'end of data' notifications shall be invoked only in state 3 in all delivery modes.

NOTE – For example, for an online service instance, the space link session has terminated, and there are no more frames to be delivered; or, for an offline service instance, all available frames between the specified start and end times (see 3.4) have been delivered.

3.7.2.4 **`notify-value`**

NOTE – The **`notify-value`** parameter is conditional on the value of `notify-type`.

3.7.2.4.1 If `notify-type` is ‘loss of frame synchronization’, then `notify-value` shall contain the UTC time when the frame synchronizer transitioned from ‘in-lock’ to ‘out-of-lock’.

3.7.2.4.2 If `notify-type` is any other value, `notify-value` shall not be present.

3.7.3 EFFECTS

3.7.3.1 Information about the occurrence of the specified event shall be delivered to the user.

3.7.3.2 The state of the provider shall remain unchanged.

3.8 RVCF-ASYNC-NOTIFY

3.8.1 PURPOSE

3.8.1.1 The RVCF service provider shall invoke the RVCF-ASYNC-NOTIFY operation to notify the user of the occurrence of an event.

NOTE – The information may be of value to the user in understanding specific provider behavior, such as an interruption in frame delivery.

3.8.1.2 RVCF-ASYNC-NOTIFY shall be confirmed by the user.

3.8.1.3 RVCF-ASYNC-NOTIFY invocations shall be delivered to the user immediately; i.e., ahead of any RVCF-TRANSFER-DATA or RVCF-SYNC-NOTIFY invocations that may be momentarily backlogged in the online frame buffer awaiting delivery.

3.8.1.4 RVCF-ASYNC-NOTIFY is valid in state 2 (‘ready’) and state 3 (‘active’) and shall be invoked only by the provider.

NOTE – Some notifications are generated only in state 3, as described below.

3.8.2 INVOCATION, RETURN, AND PARAMETERS

3.8.2.1 General

The parameters of the RVCF-ASYNC-NOTIFY operation shall be present in the invocation and return as specified in table 3-7.

Table 3-7: RVCF-ASYNC-NOTIFY Parameters

Parameters	Invocation	Return
invoker-credentials	M	
performer-credentials		M
invoke-ID	M	M
notification-type	M	
notification-value	C	
result		M
diagnostic		C

3.8.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RVCF-ASYNC-NOTIFY invocation (see 3.1.4).

3.8.2.3 `performer-credentials`

The **`performer-credentials`** parameter shall provide information that enables the invoker to authenticate the return from the performance of RVCF-ASYNC-NOTIFY (see 3.1.4).

3.8.2.4 `invoke-ID`

The performer shall return unchanged the invoker-supplied value of the `invoke-ID` parameter (see 3.1.5).

3.8.2.5 `notification-type`

The **`notification-type`** parameter shall describe the event and shall contain one of the following values:

a) 'lock status change'—the carrier demodulation process, the subcarrier demodulation process, the bit synchronization process, and/or the frame synchronization process has transitioned from 'in-lock' to 'out-of-lock' or vice versa:

1) the current lock status of carrier demodulation, subcarrier demodulation, bit synchronization, and frame synchronization shall be contained in the `notification-value` parameter (see 3.8.2.6);

2) the provider shall minimize delays in notification of these events;

NOTE – Because the determination of these lock statuses typically is based on measurements that are integrated over some time period, notification of these events may be delayed by some small amount relative to the actual occurrence of the event.

3) a lock status change notification also shall be invoked directly after a successful RVCF-START operation, before the first RVCF-TRANSFER-DATA invocation;

4) providers shall limit or suppress excessive notifications;

NOTE – Under some circumstances, an excessive number of lock status change notifications would be produced if not otherwise constrained.

5) lock status change notifications shall be invoked only in state 3 and only if the delivery mode is timely online or complete online;

b) ‘production state change’—the state of RVCF production has changed from ‘running’ to ‘halted’ or vice versa: production state change notifications shall be invoked only in state 2 or state 3 and only if the delivery mode is timely online or complete online.

NOTE – The state of RVCF production may change, for example, due to an equipment malfunction or due to a reallocation of production resources by SLE Complex Management.

3.8.2.6 `notify-value`

3.8.2.6.1 If `notify-type` is ‘lock status change’:

a) `notify-value` shall contain four lock statuses, one each for the current status of the carrier demodulation process, the subcarrier demodulation process, the bit synchronization process, and the frame synchronization process;

b) the value of each lock status shall be ‘in-lock’, ‘out-of-lock’, ‘not in use’, or ‘unknown’.

3.8.2.6.2 If `notify-type` is ‘production state change’, then `notify-value` shall contain a value indicating the current production state, which shall be ‘running’ or ‘halted’.

3.8.2.7 `result`

The parameter **`result`** shall be used to specify the result of the RVCF-ASYNC-NOTIFY operation and shall contain one of the following values:

a) ‘positive result’—the notification is accepted;

b) ‘negative result’—the notification is not accepted.

3.8.2.8 `diagnostic`

3.8.2.8.1 If `result` is ‘negative result’, `diagnostic` shall be provided and shall contain one of the following values:

a) ‘duplicate Invoke-ID’—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) ‘other’—the reason for rejection of the operation will have to be found by other means.

3.8.2.8.2 If `result` is ‘positive result’, `diagnostic` shall not be provided.

3.8.3 EFFECTS

3.8.3.1 Information about the occurrence of the specified event shall be delivered to the user.

3.8.3.2 The state of the provider shall remain unchanged.

3.9 RVCF-SCHEDULE-STATUS-REPORT

3.9.1 PURPOSE

3.9.1.1 The user shall invoke the RVCF-SCHEDULE-STATUS-REPORT operation to request that the provider send a status report either immediately or periodically.

3.9.1.2 The provider shall confirm the RVCF-SCHEDULE-STATUS-REPORT operation.

3.9.1.3 The provider shall deliver the requested status report(s) by means of the RVCF-STATUS-REPORT operation (see 3.10).

3.9.1.4 For periodic reporting, the user may change the reporting period by submitting another RVCF-SCHEDULE-STATUS-REPORT.

3.9.1.5 RVCF-SCHEDULE-STATUS-REPORT is valid in state 2 (‘ready’) and state 3 (‘active’) and shall be invoked only by the user.

3.9.2 INVOCATION, RETURN, AND PARAMETERS

3.9.2.1 General

The parameters of the RVCF-SCHEDULE-STATUS-REPORT operation shall be present in the invocation and return as specified in table 3-8.

Table 3-8: RVCF-SCHEDULE-STATUS-REPORT Parameters

Parameters	Invocation	Return
`invoker-credentials`	M	
`performer-credentials`		M
`invoke-ID`	M	M
`report-request-type`	M	
`reporting-cycle`	C	
`result`		M
`diagnostic`		C

3.9.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RVCF-SCHEDULE-STATUS-REPORT invocation (see 3.1.4).

3.9.2.3 performer-credentials

The **performer-credentials** parameter shall provide information that enables the invoker to authenticate the return from the performance of RVCF-SCHEDULE-STATUS-REPORT (see 3.1.4).

3.9.2.4 invoke-ID

The performer shall return unchanged the invoker-supplied value of the invoke-ID parameter (see 3.1.5).

3.9.2.5 report-request-type

3.9.2.5.1 The **report-request-type** parameter shall specify how reporting should be done and shall contain one of the following values:

a) ‘immediately’—send a single status report immediately;

b) ‘periodically’—send a status report every reporting-cycle seconds;

c) ‘stop’—don’t send status reports.

3.9.2.5.2 If report-request-type is ‘immediately’, the provider shall stop sending status reports after the immediate status report is sent.

NOTES

1 Periodic reporting may be restarted by means of another RVCF-SCHEDULE-STATUS-REPORT invocation.

2 The provider sends status reports by means of the RVCF-STATUS-REPORT operation.

3.9.2.6 reporting-cycle

3.9.2.6.1 If the report-request-type parameter specifies ‘periodically’, then reporting-cycle shall be present and shall specify in seconds the requested interval between status reports.

3.9.2.6.2 If the report-request-type parameter does not specify ‘periodically’, reporting-cycle shall be absent.

3.9.2.6.3 The valid range for reporting-cycle shall be set by service management.

3.9.2.7 `result`

The **`result`** parameter shall specify the result of the RVCF-SCHEDULE-STATUS-REPORT operation and shall contain on of the following values:

a) 'positive result'—the request for status reports is accepted, and the provider will send the requested report(s);

b) 'negative result'—the request for status reports is rejected, and the previous setting for status reporting remains in effect.

3.9.2.8 `diagnostic`

If `result` is 'negative result', `diagnostic` shall be provided and shall contain one of the following values:

a) 'duplicate Invoke-ID'—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) 'already stopped'—the provider is not currently providing periodic reports (when `report-request-type` is 'stop');

c) 'invalid reporting cycle'—the requested value for `reporting-cycle` is out of range;

d) 'other'—the reason for rejection of the operation will have to be found by other means.

3.9.2.8.4 If `result` is 'positive result', `diagnostic` shall not be provided.

3.9.3 EFFECTS

3.9.3.1 In the case of 'positive result':

a) if `reporting-request-type` specifies 'immediately':

   1) a status report shall be sent immediately;

   2) the sending of status reports shall then be stopped;

b) if `reporting-request-type` is 'periodically':

   1) a status report shall sent immediately;

   2) subsequent status reports shall be sent at intervals specified in `reporting-cycle`;

c) if `reporting-request-type` is ‘stop’, the sending of status reports shall be stopped.

3.9.3.2 In the case of ‘negative result’, the RVCF-SCHEDULE-STATUS-REPORT shall have no effect, and the previous setting for status reporting shall remain in effect.

3.9.3.3 The state of the provider shall remain unchanged.

3.10 RVCF-STATUS-REPORT

3.10.1 PURPOSE

3.10.1.1 The provider shall invoke the RVCF-STATUS-REPORT operation to send a status report to the user.

NOTE – RVCF-STATUS-REPORT is an unconfirmed operation.

3.10.1.2 Status reports shall be sent (or not sent) in accordance with user requests conveyed by means of the RVCF-SCHEDULE-STATUS-REPORT operation (see 3.9).

3.10.1.3 RVCF-STATUS-REPORT is valid in state 2 (‘ready’) and in state 3 (‘active’) and shall be invoked only by the provider.

3.10.2 INVOCATION, RETURN, AND PARAMETERS

3.10.2.1 General

The parameters of the RVCF-STATUS-REPORT operation shall be present in the invocation and return as specified in table 3-9.

Table 3-9: RVCF-STATUS-REPORT Parameters

Parameters	Invocation
`invoker-credentials`	M
`number-of-frames-delivered`	M

3.10.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RVCF-STATUS-REPORT invocation (see 3.1.4).

3.10.2.3 `number-of-frames-delivered`

The **`number-of-frames-delivered`** parameter shall specify the total number of telemetry frames that have been delivered to the user since the start of the service instance provision period.

3.10.3 EFFECTS

3.10.3.1 Status information shall be delivered to the user.

3.10.3.2 The state of the provider shall remain unchanged.

3.11 RVCF-GET-PARAMETER

3.11.1 PURPOSE

3.11.1.1 The user shall invoke the RVCF-GET-PARAMETER operation to ascertain the value of an RVCF service parameter.

3.11.1.2 The provider shall confirm the RVCF-GET-PARAMETER operation.

3.11.1.3 If successful, the current value of the specified RVCF service parameter shall be provided to the user in the return from the operation.

3.11.1.4 RVCF-GET-PARAMETER is valid in state 2 ('ready') and state 3 ('active') and shall be invoked only by the user.

3.11.2 INVOCATION, RETURN, AND PARAMETERS

3.11.2.1 General

The parameters of the RVCF-GET-PARAMETER operation shall be present in the invocation and return as specified in table 3-10.

Table 3-10: RVCF-GET-PARAMETER Parameters

Parameters	Invocation	Return
`invoker-credentials`	M	
`performer-credentials`		M
`invoke-ID`	M	M
`rvcf-parameter`	M	M
`parameter-value`		M
`result`		M
`diagnostic`		C

3.11.2.2 `invoker-credentials`

The **`invoker-credentials`** parameter shall provide information that enables the performer to authenticate the RVCF-GET-PARAMETER invocation (see 3.1.4).

3.11.2.3 `performer-credentials`

The **`performer-credentials`** parameter shall provide information that enables the invoker to authenticate the return from the performance of RVCF-GET-PARAMETER (see 3.1.4).

3.11.2.4 `invoke-ID`

The performer shall return unchanged the invoker-supplied value of the `invoke-ID` parameter (see 3.1.5).

3.11.2.5 `rvcf-parameter`

The **`rvcf-parameter`** parameter shall specify the RVCF service parameter whose value is to be returned to the user and shall contain one of the following values:

a) 'latency limit'—the maximum allowable delivery latency time in seconds; i.e., the maximum delay between reception of a frame on the ground and its delivery to the user: the value of this parameter in undefined if the delivery mode is not timely online;

b) 'buffer size'—the size of the online frame buffer in bytes: the value of this parameter is undefined if the delivery mode is not timely online.

3.11.2.6 `parameter-value`

The **`parameter-value`** parameter shall contain the value for the parameter indicated by `rvcf-parameter` as described in 3.11.2.5.

3.11.2.7 `result`

The **`result`** parameter shall specify the result of the RVCF-GET-PARAMETER operation and shall contain one of the following values:

a) 'positive result'—the RVCF-GET-PARAMETER operation is accepted, and the provider has returned the value of the specified RVCF service provision parameter to the user;

b) 'negative result'—the RVCF-GET-PARAMETER operation is rejected.

3.11.2.8 `diagnostic`

3.11.2.8.1 If `result` is 'negative result', `diagnostic` shall be provided and shall contain one of the following values:

a) 'duplicate Invoke-ID'—the value of the `invoke-ID` parameter is the same as the `invoke-ID` of a previous, outstanding operation;

b) 'unknown parameter'—the value of `rvcf-parameter` is not recognized by the service provider;

c) 'other'—the reason for rejection of the operation will have to be found by other means.

3.11.2.8.2 If `result` is 'positive result', `diagnostic` shall not be provided.

3.11.3 EFFECTS

3.11.3.1 If `result` is 'positive result', the value of the parameter indicated by `rvcf-parameter` shall be returned to the user.

3.11.3.2 If `result` is 'negative result', the value of the parameter indicated by `rvcf-parameter` shall not returned to the user.

3.11.3.3 The state of the provider shall remain unchanged.

3.12 RVCF-PEER-ABORT

3.12.1 PURPOSE

3.12.1.1 The RVCF-PEER-ABORT shall be invoked by an SLE provider or user to notify the peer SLE user or provider that an error has occurred and that the association between them is terminated.

NOTE – RVCF-PEER-ABORT is an unconfirmed operation.

3.12.1.2 RVCF-PEER-ABORT is valid in state 2 (‘ready’) and state 3 (‘active’) and may be invoked by either the user or the provider.

3.12.2 INVOCATION, RETURN, AND PARAMETERS

3.12.2.1 General

The parameters of the RVCF-PEER-ABORT operation shall be present in the invocation and return as specified in table 3-11.

Table 3-11: RVCF-PEER-ABORT Parameters

Parameters	Invocation
invoker-credentials	M
diagnostic	M

3.12.2.2 **invoker-credentials**

The **invoker-credentials** parameter shall provide information that enables the performer to authenticate the RVCF-PEER-ABORT invocation (see 3.1.4).

3.12.2.3 **diagnostic**

The **diagnostic** parameter shall specify why the RVCF-PEER-ABORT is issued and shall contain one of the following values:

a) ‘operational requirement’—the local system had to terminate the association to accommodate some other operational need;

b) ‘protocol error’—the local application detected an error in the sequencing of RVCF service operations;

c) ‘encoding error’—the local application detected an error in the encoding of one or more operation parameters or did not recognize the operation;

d) ‘return time-out’—the local application detected that the return from a confirmed operation was not received within a certain period of time;

e) ‘end of service provision period’—the local application detected that the service period has ended and the initiator has not invoked an UNBIND operation;

f) ‘other’—the local application detected an unspecified error during the processing of one or more operations.

3.12.3 EFFECTS

3.12.3.1 The association shall be aborted, and the user and the provider shall cease to communicate with each other.

3.12.3.2 The provider shall transition to state 1 (‘unbound’).

3.12.3.3 If the delivery mode is timely online, the provider shall discard the contents of the online frame buffer.

3.13 RVCF-PROTOCOL-ABORT

3.13.1 PURPOSE

3.13.1.1 The RVCF-PROTOCOL-ABORT operation shall notify the SLE provider or user that communications with the peer SLE user or provider have been disrupted and that the association between them is terminated.

NOTE – RVCF-PROTOCOL-ABORT is an unconfirmed operation.

3.13.1.2 An RVCF-PROTOCOL-ABORT invocation shall be conveyed to the SLE provider or user in the event that the underlying communications service signals that communications with the peer SLE user or provider have been disrupted.

3.13.1.3 RVCF-PROTOCOL-ABORT is valid in state 2 (‘ready’) and state 3 (‘active’).

3.13.2 INVOCATION, RETURN, AND PARAMETERS

3.13.2.1 General

The parameters of the RVCF-PROTOCOL-ABORT operation shall be present in the invocation and return as specified in table 3-12.

Table 3-12: RVCF-PROTOCOL-ABORT Parameters

Parameters	Invocation
`diagnostic`	M

3.13.2.2 **`diagnostic`**

The **`diagnostic`** parameter should be provided by the underlying communications service and should describe the condition that caused the RVCF-PROTOCOL-ABORT; its value is implementation dependent.

3.13.3 EFFECTS

3.13.3.1 The association shall be aborted, and the user and the provider shall cease to communicate with each other.

3.13.3.2 The provider shall transition to state 1 (‘unbound’).

3.13.3.3 The provider shall discard the contents of the online frame buffer if the delivery mode is timely online.

4 RVCF PROTOCOL

4.1 GENERIC PROTOCOL CHARACTERISTICS

4.1.1 UNEXPECTED PROTOCOL DATA UNIT

If the peer application invokes an operation not allowed in the current state of the performer, the performer shall abort the association by invoking the RVCF-PEER-ABORT operation.

4.1.2 UNDECODABLE PROTOCOL DATA UNIT

If the application receives an invocation or return that is unrecognized, contains a parameter of the wrong type, or is otherwise non-decodable, the association shall be aborted by means of the RVCF-PEER-ABORT operation.

4.1.3 MISSING RETURN

4.1.3.1 For confirmed operations, if the invoker does not receive the return from the performer within a period specified by service management, the invoker shall abort the association by invoking the RVCF-PEER-ABORT operation.

4.1.3.2 The time-out period shall be chosen taking into account performance of user and provider applications as well as the delays introduced by the underlying communications service.

4.1.3.3 The generation of the return from an operation must not depend on any human interaction.

4.1.4 FAILING AUTHENTICATION

4.1.4.1 An incoming invocation or return shall be ignored under the following conditions:

a) the credentials parameter is ‘unused’ or cannot be authenticated when, by management arrangement, credentials are required; or

b) the credentials parameter is not ‘unused’ when, by management arrangement, credentials are not required.

4.1.4.2 If an invocation is ignored, the operation shall not be performed.

4.1.4.3 If a return is ignored, it shall be as if no report of the outcome of the operation has been received.

4.1.5 CONCURRENT PROCESSING

4.1.5.1 After invoking an operation, the invoker may invoke another operation without waiting for the return from the first invocation.

NOTES

1 The `invoke-ID` parameter allows the invoker to associate a return with the corresponding invocation.

2 RVCF-BIND and RVCF-UNBIND are exceptions: the invoker must always block on RVCF-BIND; and no additional operations may be invoked after RVCF-UNBIND.

4.1.5.2 The invoker may opt not to exploit the non-blocking capability and always wait for the return before invoking the next operation.

4.1.5.3 In that case, the `invoke-ID` parameter may be set to 'null'.

4.2 STATE TRANSITION TABLE

NOTE – The state table describes service operation interactions and state transitions for the service provider in its role as either initiator or responder. The structure of the state table is:

- The leftmost column lists all incoming events. Where these events correspond to the arrival of an incoming SLE-PDU, in addition to a 'verbal' description of the event, the ASN.1 type defined for this SLE-PDU in Annex A is indicated in parentheses ().

- Where an event is internal to the provider, its description is put in quotation marks ' '. A cross reference to further description of the event is given in the notes below the table.

- Where an event is triggered by an unrecognized SLE-PDU, i.e. when the ASN.1 type of the SLE-PDU is unknown ('unknown SLE-PDU') or the SLE-PDU fails to authenticate ('not authenticated SLE-PDU'), its description is put in quotation marks.

- The three columns (one column per state) on the right side of the table specify the behavior the provider will exhibit, which depends on the current state and the incoming event. In some cases, the behavior in addition depends on Boolean conditions, also referred to as predicates. Such conditions are put in double quotes " ". The dependency is presented in form of an IF <condition> THEN <action> [ELSEIF <condition> THEN <action>] ELSE <action> clause.

- If the action is simply to send a specific SLE-PDU, that is indicated by the ASN.1 type of this SLE-PDU in parentheses ( ). The convention has been introduced to use ‘+’ and ‘-’ to represent positive and negative responses respectively. Where several actions are to be taken, this is indicated by a ‘compound action’ which is put in curly braces { }. The individual actions making up this compound action are identified in the notes below the table.

- ‘Not applicable’ is stated where the given event can only occur in the given state because of an implementation error on the provider side.

- Where the consequences of an incoming event are not visible to the user because the provider does not send any SLE-PDU in reaction to the given event, the action is put in square brackets [].

- State transitions are indicated by an arrow and the number of the state, which will be entered: for example, → 1 indicates the transition to state 1. This notation has also been used to show no change in state.

- The actions to be taken and the state transition are considered to be atomic operations. The sequence is irrelevant except that SLE-PDUs shall be sent in the sequence stated in the table.

Table 4-1: Provider Behavior

Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
Negative BIND return (-rvcfBindReturn) * provider initiated BIND	[ignore] → 1	{peer abort} → 1	{peer abort} → 1
Positive BIND return (+rvcfBindReturn) * provider initiated BIND	[ignore] → 2	{peer abort} → 1	{peer abort} → 1

BIND invocation
(rvcfBindInvoke)
 	
* user initiated BIND
	IF “positive result”
	THEN → 2
	(+rvcfBindReturn)
	ELSE → 1
	(-rvcfBindReturn)
		{peer abort} → 1	{peer abort} → 1

UNBIND return (rvcfUnbindReturn) * provider initiated UNBIND	[ignore] → 1	stop reporting timer → 1 stop all confirmation timers	{peer abort} → 1
UNBIND invocation (rvcfUnbindInvoke) * user initiated UNBIND	[ignore] → 1	stop reporting timer → 1 stop all confirmation timers, (+rvcfUnbindResponse)	{peer abort} → 1

START invocation
(rvcfStartInvoke)
	[ignore] → 1	IF “positive result”
		THEN → 3
		{start transferring data}
		ELSE → 2
		(-rvcfStartResponse)
			{peer abort} → 1

Table 4-1: Provider Behavior (continued)

Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
STOP invocation (rvcfStopInvoke)	[ignore] → 1	{peer abort} → 1	{stop transferring data} → 2
‘data available’ event	Not applicable	Not applicable	send content of the buffer → 3

SCHEDULE-STATUS-REPORT invocation
(rvcfScheduleStatusReportInvoke)
	[ignore] → 1	IF “immediate report”
		THEN → 2
		{immediate report}
		ELSE IF “periodic report”
		THEN → 2
		{periodic report}
		ELSE IF “stop reporting”
		THEN → 2
		{stop reporting}
		ELSE → 2
		(-rvcfScheduleStatusReportReturn)
			IF “immediate report”
			THEN → 3
			{immediate report}
			ELSE IF “periodic report”
			THEN → 3
			{periodic report}
			ELSE IF “stop reporting”
			THEN → 3
			{stop reporting}
			ELSE → 3
			(-rvcfScheduleStatusReportReturn)

‘reporting timer expired’ event	Not applicable	(rvcfStatusReportInvocation) → 2	(rvcfStatusReportInvocation) → 3
‘confirmation <n> timer expired’ event	Not applicable	{peer abort} → 1	{peer abort} → 1

GET-PARAMETER invocation
(rvcfGetParameterInvoke)
	[ignore] → 1	IF “positive result”
		THEN → 2
		(+rvcfGetParameterReturn)
		ELSE → 2
		(-rvcfGetParameterReturn)
			IF “positive result”
			THEN → 3
			(+rvcfGetParameterReturn)
			ELSE → 3
			(-rvcfGetParameterReturn)

‘RF carrier lock status change’ or ‘Subcarrier lock status change’ or ‘Bit sync lock status change’ or ‘Frame sync lock status changes’	Not applicable	{notify} → 2	{notify} → 3

Table 4-1: Provider Behavior (continued)

Incoming Event	Unbound (State 1)	Ready (State 2)	Active (State 3)
‘end of data’ event	Not applicable	Not applicable	{notify} → 3
‘production halted’ event	Not applicable	{notify} → 2	{notify} → 3
‘production ready’ event	Not applicable	{notify} → 2	{notify} → 3
ASYNCHRONOUS-NOTIFY return (rvcfAsynchronousNotifyReturn)	[ignore] → 1	stop confirmation <n> timer → 2	stop confirmation <n> timer → 3
PEER ABORT (rvcfPeerAbortInvoke)	[ignore] → 1	{clean up} → 1	{clean up} → 1
PROTOCOL ABORT (rvcfProtocolAbortInvoke)	[ignore] → 1	{clean up} → 1	{clean up} → 1
‘unknown PDU’ event	[ignore] → 1	{peer abort} → 1	{peer abort} → 1
‘not authenticated PDU’ event	[ignore] → 1	[ignore] → 2	[ignore] → 3

NOTES:

1 “positive result” evaluates to TRUE if all checks on the invocation are passed.

2 {peer abort} covers the following actions:

a) invoke (RvcfPeerAbortInvoke);

b) stop status reporting;

c) stop all return timers;

d) stop latency-limit timer;

e) clear online frame buffer if delivery mode is timely online.

3 “return <n> pending” evaluates to TRUE if the provider has invoked a confirmed operation with `invoke-ID` set to <n> and the return from that operation has not been received. Since the interface may be operated in a non-blocking mode, several returns may be pending at any given point in time, even for the same type of operation. Whenever the provider invokes a confirmed operation with `invoke-ID` set to <n>, it shall start an associated return <n> timer. Should this timer expire before the return <n> is received, the provider shall invoke RVCF-PEER-ABORT.

4 “immediate report” evaluates to TRUE if all parameter checks on the invocation are passed and if the invocation requests an immediate status report.

5 {immediate report} covers the following actions:

a) send (+RvcfScheduleStatusReportReturn);

b) invoke (RvcfStatusReport).

6 “periodic report” evaluates to TRUE if all parameter checks on the invocation are passed and if the invocation requests periodic status reports.

7 {periodic report} covers the following actions:

a) send (+RvcfScheduleStatusReportReturn);

b) invoke (RvcfStatusReport);

c) start reporting timer (and send additional reports each reporting period).

8 “stop reporting” evaluates to TRUE if all parameter checks on the invocation are passed and the if the invocation requests periodic status reporting to be stopped.

9 {stop reporting} covers the following actions:

a) send (+RvcfScheduleStatusReportReturn);

b) stop reporting timer.

10 {report} covers the following action:

a) buffer (RvcfSynchronousNotifyInvocation); or

b) send (RvcfAsynchronousNotifyInvocation).

11 {notify} covers the following action: buffer (RvcfSynchronousNotifyInvocation)

12 ‘start transferring data’ covers the following actions:

a) send (+RvcfStartReturn);

b) set latency timer;

c) start buffering.

13 ‘stop transferring data’ covers the following actions:

a) stop sending (RvcfTransferDataInvoke);

b) notify ‘end of data’ event;

c) stop buffering if timely online;

d) clear buffer if timely online;

e) stop latency timer;

f) send (+RvcfStopReturn).

14 {clean up} covers the following actions:

a) stop all reporting timers;

b) stop all confirmation timers;

c) stop latency timer;

d) clear buffer if timely online.

15 ‘data available’ correspond to

a) latency timer expired and buffer not empty; or

b) buffer is full.

16 Further descriptions of events can be found at the following locations.

Event	Reference
‘RF carrier lock status change’	3.8.2.5
‘Subcarrier lock status change’	3.8.2.5
‘Bit sync lock status change’	3.8.2.5
‘Frame sync status change’	3.8.2.5
‘change applied’	3.7.2.3
‘end of data’	3.7.2.3
‘unknown PDU’	0
‘not authenticated PDU’	4.1.4
‘production halted’	3.8.2.5
‘production ready’	3.8.2.5
‘reporting timer expired’	3.9.2.6
‘confirmation <n> timer expired’	note 3 above
‘data available’	note 15 above

ANNEX A

DATA TYPE DEFINITIONS

(This annex **is** part of the Recommendation)

A1 INTRODUCTION

This annex defines the data types that are used by the RVCF service. It is intended to provide a clear specification of these data types and to avoid ambiguity. It is not intended to constrain how these data types are implemented or encoded. These definitions are suitable for inclusion in any type of ASN.1 based protocol that implements the RVCF service.

The data type definitions are presented in five ASN.1 modules.

Subsection A2.1 contains basic types intended be common with other SLE Transfer Services. As more services become specified by CCSDS, further types may be added to this module or existing types may be extended. This does, however, not invalidate the module in its present form because an implementation compliant with a future extended version of this module is still fully interoperable with an implementation based on its present version.

Subsection A2.2 specifies those SLE-PDUs associated with binding and unbinding that are exchanged between an SLE user and an SLE provider that are not specific to one SLE transfer service type.

Subsection A2.3 specifies those SLE-PDUs associated with SLE service operations that are not specific for one SLE transfer service type.

Subsection A2.4 specifies data types considered specific for the RVCF service. In part, these specific types are derived from types specified in A2.1 by means of subtyping.

Subsection A2.5 specifies all incoming (from a provider point of view) SLE-PDUs. Where applicable, these SLE-PDUs are mapped to the generic SLE-PDUs defined in A2.2 and A2.3.

Subsection A2.6 specifies in the same way the outgoing SLE-PDUs.

ASN.1 would have permitted to formulate several SLE-PDUs in a more generic way and to introduce the SLE transfer service type specific elements by means of Information Objects. However, it does not provide any added value, since the SLE transfer services call for the definition of an application layer protocol which as such has to be fully specified without leaving any types open for later specification by the user of this protocol. Furthermore, support by ASN.1 tools for more recent features like information objects is still somewhat limited. The specification using such features is less comprehensible for non-experts in the field, while the ASN.1 elements used in this Recommendation are deemed to be fairly intuitive.

The exception to what is stated above occurs in two BIND parameters. These parameters are specified in module A2.4: `Port-Identifier` and `Application-Context`. The internal structure of these two parameters is determined by the requirements of the underlying telecommunications provider. As new technologies keep emerging, a type specification is not likely to be exhaustive. In order to better illustrate the purpose of these parameters, module A2.4 presents a type specification which accommodates the needs of a full ISO/OSI protocol suite and of DCE. For other technologies, an OCTET STRING is permitted. It should be noted, however, that interoperability between two SLE transfer service applications is only accomplished, when both applications apply the same internal structuring of this OCTET STRING.

In the interest of backward compatibility of a new version of an SLE transfer service with a previous version, types which could have been specified as ENUMERATED, have rather been mapped to named INTEGER. Addition of further labels associated with INTEGER values not yet used does not create a new type. Extension of an ENUMERATED type, however, actually creates a new type. An application version not yet upgraded to the new ASN.1 definition would not even be able to decode a PDU containing such ENUMERATED, even though the additional features of a new application version are not used.

Although A2.4 and A2.5 define the SLE-PDUs which will be exchanged between the SLE provider and user applications, they shall not be understood to require that these SLE-PDUs shall completely be mapped to the user data field of the underlying communications protocol. Depending on the communications service used, SLE-PDUs may be profiled in a different manner, provided that equivalent functionality is achieved.

In order to support navigation within the modules, the elements are presented in alphabetic order. Furthermore, those elements which are referenced in another module (i.e., are imported into that module) are printed in bold face in the module where they are defined.

A 2 RVCF DATA TYPE SPECIFICATION

A 2.1 SLE TRANSFER SERVICE COMMON TYPES

```
CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

EXPORTS     ConditionalTime
,           Credentials
,           Diagnostics
,           Duration
,           ForwardDuStatus
,           IntPosLong
,           IntPosShort
,           IntUnsignedLong
,           IntUnsignedShort
,           ParameterName
,           ProductionStatus
,           Report
,           InvokeId
,           SpaceLinkDataUnit
,           Time
;

ConditionalTime              ::=   CHOICE
{     undefined              [0]   NULL
,     known                  [1]   Time
}

Credentials                  ::=   CHOICE
{     unused                 [0]   NULL
,     used                   [1]   BIT STRING (SIZE (64 .. 2048))
}

Diagnostics                  ::=   INTEGER
{     duplicateInvokeId      (100)
,     otherReason            (101)
}

-- The Duration is expressed in microseconds
Duration                     ::= IntUnsignedLong

ForwardDuStatus              ::=   INTEGER
{     radiated               (0)
,     expired                (1)
,     interrupted            (2)
,     transferred            (3)
-- FSP
,     productionStarted      (4)
,     undefined              (5)
-- FSP
}
```

```
-- 1 to (2^32)-1
IntPosLong                    ::=   INTEGER (1 .. 42949667295)

-- 1 to (2^16)-1
IntPosShort                   ::=   INTEGER (1 .. 65535)

-- 0 to (2^32)-1
IntUnsignedLong               ::=   INTEGER (0 .. 42949667295)

-- 0 to (2^16)-1
IntUnsignedShort              ::=   INTEGER (0 .. 65535)

InvokeId                      ::=   IntUnsignedShort

ParameterName                 ::=   INTEGER
{       aggregation-usage           (0)
,       buffer-size                 (1)
,       cop-in-effect               (2)
,       directive-identification    (3)
,       fecf-usage                  (4)
,       fop-sliding-window          (5)
,       fop-state                   (6)
,       frame-Quality               (7)
,       latency-Limit               (8)
,       map-mux-scheme              (9)
,       map-mux-control             (10)
,       maximum-packet-size         (11)
,       maximum-segment-size        (12)
,       maximum-frame-size          (13)
,       segmentation                (14)
,       sent-queue-size             (15)
,       transmission-limit          (16)
,       timer-initial               (17)
,       timeout-type                (18)
,       vc-mux-scheme               (19)
,       vc-mux-control              (20)
,       virtual-channel             (21)
}

ProductionStatus              ::=   INTEGER
{     halted                  (0)
,     waiting                 (1)
,     running                 (2)
}

Report                        ::=   INTEGER
{     produceReport           (0)
,     doNotProduceReport      (1)
}

SpaceLinkDataUnit             ::=   OCTET STRING (SIZE (7 .. 65536))

Time                          ::=   CHOICE
{     ccsdsFormat             [0]   TimeCCSDS
}
```

```
TimeCCSDS                     ::=   OCTET STRING (SIZE(8))
-- P-field is implicit (not present, defaulted to 41 hex
-- T-field:
-- 2 octets: number of days since 1958/01/01 00:00:00
-- 4 octets: number of milliseconds of the day
-- 2 octets: number of microseconds of the millisecond
--           (set to 0 if not used)
-- This definition reflects exactly the format of the CCSDS defined
-- time tag as used in spacelink data units.

END
```

A 2.2 SLE TRANSFER SERVICE BIND TYPES

```
CCSDS-SLE-TRANSFER-SERVICE-BIND-TYPES

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

EXPORTS     Sle-Bind-Invocation
,           Sle-Bind-Return
,           Sle-Peer-Abort
,           Sle-Protocol-Abort
,           Sle-Unbind-Invocation
,           Sle-Unbind-Return
;

IMPORTS     Credentials
,           IntPosShort
,           IntUnsignedShort
      FROM  CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES
;

-- =============================================================
-- The first part of the module definition contains the SLE-PDUs
-- =============================================================

Sle-Bind-Invocation           ::=   SEQUENCE
{     invokerCredentials            Credentials
,     initiatorPortIdentifier       PortId
,     responderPortIdentifier       PortId
,     serviceType                   ServiceType
,     versionNumber                 VersionNumber
,     serviceInstanceIdentifier     ServiceInstanceIdentifier
,     applicationContext            ApplicationContext
}

Sle-Bind-Return               ::=   SEQUENCE
{     performerCredentials          Credentials
,     result                  CHOICE
      {     positive          [2]   SEQUENCE
            {     versionNumber           VersionNumber
            ,     applicationContext      ApplicationContext
            }
      ,     negative          [3]   BindDiagnostic
      }
}

Sle-Peer-Abort                ::=   SEQUENCE
{     invokerCredentials            Credentials
,     diagnostic                    PeerAbortDiagnostic
}

Sle-Protocol-Abort            ::=   ProtocolAbortDiagnostic

Sle-Unbind-Invocation         ::=   SEQUENCE
{     invokerCredentials            Credentials
,     unbindReason                  UnbindReason
}
```

```
Sle-Unbind-Return             ::=   Credentials

-- ==============================================================
-- The second part of the module definition contains the types
-- used by the SLE-PDUs declared in the first part.
-- ==============================================================

ApplicationContext            ::=   CHOICE
{     osiApplicationContext         [0]   Context
,     dceApplicationContext         [1]   OCTET STRING (SIZE (0 .. 128))
,     genericApplicationContext     [2]   OCTET STRING (SIZE (1 .. 128))
,     fixedApplicationContext       [3]   NULL
}

ApplicationIdentifier         ::=   INTEGER
{     rtnAllFrames            (0)
,     rtnInsert               (1)
,     rtnMcFrames             (2)
,     rtnVcFrames             (3)
,     rtnVcFsh                (4)
,     rtnVcOcf                (5)
,     rtnMcFsh                (6)
,     rtnMcOcf                (7)
,     rtnBitstr               (8)
,     rtnAosSpacePkt          (9)
,     fwdSpacePkt             (10)
,     fwdVca                  (11)
,     fwdBitstr               (12)
,     fwdProtoVcdu            (13)
,     fwdInsert               (14)
,     fwdCVcdu                (15)
,     fwdTcSpacePkt           (16)
,     fwdTcVca                (17)
,     fwdTcFrame              (18)
,     fwdCltu                 (19)
}

-- This type uses the ANY DEFINED BY feature.
-- That feature is not supported in the 94 ASN.1 !!
AttributeValueAssertion       ::=   SEQUENCE
{     attributeId             OBJECT IDENTIFIER
,     attributeValue          ANY DEFINED BY attributeId
}

BindDiagnostic                ::=   INTEGER
{     noSuchServiceInstance               (0)
,     invalidTime                         (1)
,     unableToComply                      (2)
,     inconsistentServiceType             (3)
,     versionNotSupported                 (4)
,     applicationContextNotSupported      (5)
,     otherReason                         (1000)
}

Context                       ::=   SEQUENCE
{     serviceSyntax                       Syntax
,     supplementaryInformationSyntax      Syntax
,     telecommunicationServices           TelecomService
}
```

```
DcePort                       ::=   SEQUENCE
{     ipAddress   CHOICE
            {ipv4 [0]   OCTET STRING (SIZE (4))
            ,ipv6 [1]   OCTET STRING (SIZE (16))
            }
,     endPoint    OCTET STRING (SIZE (1 .. 128))
}

GenericPort                   ::=   OCTET STRING (SIZE (1..128))

OsiAddress                    ::=   SEQUENCE
{     presentationSelector    OCTET STRING (SIZE (0 .. 4))
,     sessionSelector         OCTET STRING (SIZE (0 .. 16))
,     transportSelector       OCTET STRING (SIZE (0 .. 16))
,     networkSelector         OCTET STRING (SIZE (0 .. 16))
}

PeerAbortDiagnostic           ::=   INTEGER
{     operationalRequirement              (0)
,     protocolError                       (1)
,     pduEncodingError                    (2)
,     returnTimeout                       (3)
,     endOfServiceProvisionPeriod         (4)
,     otherReason                         (1000)
}

PortId                        ::=   CHOICE
{     osiInitiatorPort        [0]   OsiAddress
,     dceInitiatorPort        [1]   DcePort
,     genericInitiatorPort    [2]   GenericPort
}
-- A given implementation only needs to implement the
-- choice(s) corresponding to the underlying communications
-- technologies actually used by this implementation.

ProtocolAbortDiagnostic       ::=   CHOICE
{     localForm               [0]   OCTET STRING (SIZE (1 .. 128))
}

RDNSequence                   ::=   SEQUENCE OF
                        RelativeDistinguishedName     (SIZE (5 .. 256))

RelativeDistinguishedName     ::=   SET SIZE (1 .. MAX) OF
                        AttributeValueAssertion

SessionRequirements           ::=   OCTET STRING (SIZE ( 0 .. 256 ))

ServiceInstanceIdentifier     ::=   RDNSequence

ServiceType                   ::=   CHOICE
{     osiApplication          [0]   OBJECT IDENTIFIER
,     ipApplication           [1]   ApplicationIdentifier
,     genericApplication      [2]   OCTET STRING (SIZE (1 .. 128))
}
-- A given implementation only needs to implement the
-- choice(s) corresponding to the underlying communications
-- technologies actually used by this implementation.
```

```
Syntax                        ::=   SEQUENCE
{     syntaxID                IntUnsignedShort
,     abstractSyntax          OBJECT IDENTIFIER
,     transferSyntax          OBJECT IDENTIFIER
}

TelecomService                ::=   SEQUENCE
{     sessionRequ             SessionRequirements
}

UnbindReason                  ::=   INTEGER
{     end                     (0)
,     suspend                 (1)
,     versionNotSupported     (2)
,     other                   (1000)
}

VersionNumber                 ::=   IntPosShort

END
```

A 2.3 SLE TRANSFER COMMON PDUs

```
CCSDS-SLE-TRANSFER-COMMON-PDUS

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

EXPORTS     Sle-Acknowledgement
,           Sle-Schedule-Status-Report-Invocation
,           Sle-Schedule-Status-Report-Return
,           Sle-Stop-Invocation
;

IMPORTS     Credentials
,           Diagnostics
,           IntUnsignedShort
,           InvokeId
      FROM  CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES
;

-- ================================================================
-- The first part of the module definition contains the SLE-PDUs
-- ================================================================
Sle-Acknowledgement           ::=   SEQUENCE
{     credentials             Credentials
,     invokeId                InvokeId
,     result                  CHOICE
      {     positiveResult    [0]   NULL
      ,     negativeResult    [1]   Diagnostics
      }
}

Sle-Schedule-Status-Report-Invocation     ::=   SEQUENCE
{     userCredentials                     Credentials
,     invokeId                            InvokeId
,     reportRequestType                   ReportRequestType
}

Sle-Schedule-Status-Report-Return   ::=   SEQUENCE
{     performerCredentials          Credentials
,     invokeId                      InvokeId
,     result                        CHOICE
      {     positiveResult    [0]   NULL
      ,     negativeResult    [1]   DiagnosticScheduleStatusReport
      }
}

Sle-Stop-Invocation           ::=   SEQUENCE
{     userCredential          Credentials
,     invokeId                InvokeId
}
```

```
-- =============================================================
-- The second part of the module definition contains the types
-- used by the SLE-PDUs declared in the first part.
-- =============================================================

DiagnosticScheduleStatusReport      ::=   CHOICE
{     common                  [0]   Diagnostics
,     specific                [1]   INTEGER
      {     alreadyStopped          (0)
      ,     invalidReportingCycle   (1)
      }
}

-- The cycle duration is expressed in seconds
ReportingCycle                ::=   INTEGER (60 .. 600)

ReportRequestType             ::=   CHOICE
{     immediately             [0]   NULL
,     periodically            [1]   ReportingCycle
,     stop                    [2]   NULL
}

END
```

A 2.4 SLE TRANSFER RVCF STRUCTURES

```
CCSDS-SLE-TRANSFER-RVCF-STRUCTURES

DEFINITIONS
IMPLICIT TAGS
::=  BEGIN

EXPORTS     AntennaId
,           AsyncNotification
,           DiagnosticRvcfGet
,           DiagnosticRvcfStart
,           GVcId
,           LockStatus
,           RvcfGetParameter
,           SyncNotification

IMPORTS     Diagnostics
,           IntPosLong
,           IntPosShort
,           ParameterName
,           ProductionStatus
,           Time
      FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES

AntennaId               ::=   CHOICE
{     globalForm        [0]   OBJECT IDENTIFIER
,     localForm         [1]   OCTET STRING (SIZE (1 .. 16))
}

AsyncNotification       ::=   CHOICE
{     lockStatusChange        [0]   LockStatuses
,     productionStatusChange  [1]   ProductionStatusChange
}

DiagnosticRvcfGet       ::=   CHOICE
{     common            [0]   Diagnostics
,     specific          [1]   INTEGER
      {     unknownParameter  (0)
      }
}


DiagnosticRvcfStart     ::=   CHOICE
{     common            [0]   Diagnostics
,     specific          [1]   INTEGER
      {     unableToComply    (0)
      ,     invalidStartTime  (1)
      ,     invalidStopTime   (2)
      ,     missingTimeValue  (3)
      ,     invalidGVcId      (4)
      }
}
```

```
GVcId                   ::=   SEQUENCE
{     spacecraftId      INTEGER (0 .. 1023)
,     versionNumber     INTEGER (0 .. 3)
,     vcId              INTEGER (0 .. 63)
}
-- Note that the permissible range of the vcId depends on the
-- versionNumber.

LockStatus              ::=   INTEGER
{     inLock      (0)
,     outOfLock   (1)
,     notInUse    (2)
,     unknown     (3)
}

LockStatuses            ::=   SEQUENCE
{     rfLock            LockStatus
,     subcarrierLock    LockStatus
,     bitLock           LockStatus
,     frameLock         LockStatus
}

ProductionStatusChange  ::=   ProductionStatus
(     running
|     halted
)

RvcfGetParameter        ::=   CHOICE
{     parLatencyLimit   [0]   SEQUENCE
      {     parameterName     ParameterName (latency-limit)
      ,     parameterValue    IntPosShort
      }
,     parBufferSize     [1]   SEQUENCE
      {     parameterName     ParameterName (buffer-size)
      ,     parameterValue    IntPosLong
      }
}

SyncNotification        ::=   CHOICE
{     lossFrameSync     [0]   Time
,     dataDiscarded     [1]   NULL
,     endOfData         [2]   NULL
}

END
```

A 2.5 RVCF INCOMING PDUs

```
CCSDS-SLE-TRANSFER-RVCF-INCOMING-PDUS

DEFINITIONS
IMPLICIT TAGS
::=  BEGIN

IMPORTS     Credentials
,           InvokeId
,           ParameterName
,           Time
      FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES

            Sle-Stop-Invocation
,           Sle-Acknowledgement
,           Sle-Schedule-Status-Report-Invocation
      FROM CCSDS-SLE-TRANSFER-COMMON-PDUS

            GVcId
,           RvcfGetParameter
      FROM CCSDS-SLE-TRANSFER-RVCF-STRUCTURES

            Sle-Bind-Invocation
,           Sle-Bind-Return
,           Sle-Peer-Abort
,           Sle-Protocol-Abort
,           Sle-Unbind-Invocation
,           Sle-Unbind-Return
      FROM CCSDS-SLE-TRANSFER-SERVICE-BIND-TYPES

-- =============================================================
-- The first part of the module definition contains the RVCF type
-- that contains all the possible PDUs the provider may receive.
-- =============================================================

Rvcf-User-to-Provider-Pdu           ::=   CHOICE
{     rvcf-Bind-Invocation          [100] Sle-Bind-Invocation
,     rvcf-Bind-Return              [101] Sle-Bind-Return
,     rvcf-Unbind-Invocation        [102] Sle-Unbind-Invocation
,     rvcf-Unbind-Return            [103] Sle-Unbind-Return
,     rvcf-Start-Invocation         [0]   RvcfStartInvocation
,     rvcf-Stop-Invocation          [1]   Sle-Stop-Invocation
,     rvcf-Async-Notify-Return      [2]   Sle-Acknowledgement
,     rvcf-Schedule-Status-Report-Invocation    [3]
                              Sle-Schedule-Status-Report-Invocation
,     rvcf-Get-Parameter-Invocation [4]   RvcfGetParameterInvocation
,     rvcf-Peer-Abort-Invocation    [104] Sle-Peer-Abort
-- The protocol abort PDU can only be issued by the
-- communication and is received by the SLE provider.
,     rvcf-Protocol-Abort-Invocation      [105] Sle-Protocol-Abort
}
```

```
-- ============================================================
-- The second part of the module definition contains the types
-- used by the RVCF-PDUs declared in the first part.
-- ============================================================

RvcfGetParameterInvocation    ::=   SEQUENCE
{     invokerCredentials      Credentials
,     invokeId                InvokeId
,     rvcfParameter           ParameterName
                                    (latency-limit
                                    |buffer-size
                                    )
}

RvcfStartInvocation           ::=   SEQUENCE
{     invokerCredentials      Credentials
,     invokeId                InvokeId
,     startTime               Time
,     stopTime                Time
,     globalVcId              GVcId
}

END
```

A 2.6 RVCF OUTGOING PDUS

```
CCSDS-SLE-TRANSFER-RVCF-OUTGOING-PDUS

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

IMPORTS     Credentials
,           IntUnsignedLong
,           IntUnsignedShort
,           InvokeId
,           SpaceLinkDataUnit
,           Time
      FROM CCSDS-SLE-TRANSFER-SERVICE-COMMON-TYPES

            Sle-Acknowledgement
,           Sle-Schedule-Status-Report-Return
      FROM CCSDS-SLE-TRANSFER-COMMON-PDUS

            AntennaId
,           AsyncNotification
,           DiagnosticRvcfGet
,           DiagnosticRvcfStart
,           RvcfGetParameter
,           SyncNotification
      FROM CCSDS-SLE-TRANSFER-RVCF-STRUCTURES

            Sle-Bind-Invocation
,           Sle-Bind-Return
,           Sle-Peer-Abort
,           Sle-Protocol-Abort
,           Sle-Unbind-Invocation
,           Sle-Unbind-Return
      FROM CCSDS-SLE-TRANSFER-SERVICE-BIND-TYPES

-- =============================================================
-- The first part of the module definition contains the RVCF type
-- that contains all the possible PDUs the provider may send.
-- =============================================================

Rvcf-Provider-to-User-Pdu           ::=   CHOICE
{     rvcf-Bind-Invocation          [100] Sle-Bind-Invocation
,     rvcf-Bind-Return              [101] Sle-Bind-Return
,     rvcf-Unbind-Invocation        [102] Sle-Unbind-Invocation
,     rvcf-Unbind-Return            [103] Sle-Unbind-Return
,     rvcf-Start-Return             [0]   RvcfStartReturn
,     rvcf-Stop-Return              [1]   Sle-Acknowledgement
,     rvcf-Transfer-Data-Invocation [2]   RvcfTransferDataInvocation
,     rvcf-Sync-Notify-Invocation   [3]   RvcfSyncNotifyInvocation
,     rvcf-Async-Notify-Invocation  [4]   RvcfAsyncNotifyInvocation
,     rvcf-Schedule-Status-Report-Return       [5]
                                    Sle-Schedule-Status-Report-Return
```

```
,     rvcf-Status-Report-Invocation [6]   RvcfStatusReportInvocation
,     rvcf-Get-Parameter-Return     [7]   RvcfGetParameterReturn
,     rvcf-Peer-Abort-Invocation    [104] Sle-Peer-Abort
-- The protocol abort PDU can only be issued by the
-- communication and is received by the SLE user.
,     rvcf-Protocol-Abort-Invocation      [105] Sle-Protocol-Abort
}

-- =============================================================
-- The second part of the module definition contains the types
-- used by the RVCF-PDUs declared in the first part.
-- =============================================================

RvcfAsyncNotifyInvocation     ::=   SEQUENCE
{     invokerCredentials      Credentials
,     invokeId                InvokeId
,     notification            AsyncNotification
}

RvcfGetParameterReturn        ::=  SEQUENCE
{     performerCredentials    Credentials
,     invokeId                InvokeId
,     rvcfGetParameter        RvcfGetParameter
,     result                  CHOICE
     {      positiveResult    [0]   NULL
     ,      negativeResult    [1]   DiagnosticRvcfGet
     }
}

RvcfStartReturn               ::=   SEQUENCE
{performerCredentials         Credentials
,     invokeId                InvokeId
,     result                  CHOICE
     {      positiveResult    [0]   NULL
     ,      negativeResult    [1]   DiagnosticRvcfStart
     }
}

RvcfStatusReportInvocation    ::=   SEQUENCE
{     invokerCredentials      Credentials
,     invokeId                InvokeId
,     deliveredFrameNumber    IntUnsignedLong
}

RvcfSyncNotifyInvocation      ::=   SEQUENCE
{     invokerCredentials      Credentials
,     notification            SyncNotification
}

RvcfTransferDataInvocation    ::=   SEQUENCE
{     invokerCredentials      Credentials
,     earthReceiveTime        Time
,     antennaId               AntennaId
,     dataLinkContinuity      IntUnsignedShort
,     privateAnnotation       CHOICE
      {     null              [2]   NULL
      ,     notNull           [1]   OCTET STRING (SIZE ( 1 ..128))
      }
,     data                    SpaceLinkDataUnit
}

END
```

ANNEX B

INDEX TO DEFINITIONS

(This annex **is not** part of the Recommendation)

TBS.

ANNEX C

ACRONYMS

(This annex **is not** part of the Recommendation)

This annex lists the acronyms used in this Recommendation.

TBS

ANNEX D

CONFORMANCE MATRIX

(This annex **is not** part of the Recommendation)

This annex provides the Conformance Matrix for implementations of the RVCF service. An implementation of the RVCF service will be considered to be 'conformant' if the mandatory elements identified in the Tables D-1 and D-2 are implemented as described in this Recommendation.

Table D-1: Conformance Matrix for RVCF Service (Operations)

RVCF Service Operation	Optional/Mandatory
RVCF-BIND RVCF-UNBIND	These operations are mandatory. All parameters are also mandatory with the exception of: Invoker-credentials Performer-credentials
RVCF-START RVCF-STOP RVCF-TRANSFER-DATA RVCF-SYNC-NOTIFY RVCF-SCHEDULE-STATUS-REPORT RVCF-STATUS-REPORT RVCF-SYNC-SET-PARAMETER RVCF-GET-PARAMETER	These operations are mandatory. All parameters are also mandatory with the exception of: Invoker-credentials Performer-credentials
RVCF-PEER-ABORT	This operation is mandatory. All parameters are also mandatory with the exception of: Invoker-credentials
RVCF-PROTOCOL-ABORT	This operation and its parameter are mandatory.

Table D-2: Conformance Matrix for RVCF Service (Other Requirements)

Requirement	Optional / Mandatory
Online buffer size (3.1.7)	Mandatory minimum size is 1024 maximum-sized telemetry frames
State transition table (4.2)	Mandatory
ASN.1 protocol specification (annex A)	All data types are mandatory. Implementations may vary as described in the annex.